računari

u vašoj kući

Amstrad • Spectrum

Commodore 64 • Electron

3

Izdaje BIGZ OOUR „Duga“

Specijalno izdanje časopisa „Galaksija

Decembar 1984.

Cena 200 D

BBC B • galaksija

disk jedinice „amstrad“

„spektrum plus“

neki bolji bejzici

program

od milion dolara

kompletna škola

avanturističkih igara

pedeset veličanstvenih

ilustrovani katalog

najboljih igara

za „spektrum“

poseban prilog

ROM 2

za „galaksiju“

U 1985. god. POJAVICE SE NA TRŽIŠTU

KUĆNI RAČUNAR PECOM 32

OBRAZOVANJE ★ ZABAVA ★ VOĐENJE POSLOVANJA ★ REŠAVANJE MATEMATIČKIH PROBLEMA ★ VOĐENJE KUĆNIH POSLOVA

KARAKTERISTIKE

CPU	CDP 1802B (5MHz, 5V7)
ROM	16 KB (12 KB za BASIC 3,4 KB za sistemski softver)
STANDARDNI RAM	36 KB
KORISNIČKI RAM	32 KB
SOFTVER	BASIC 3 (rad na mašinskom jeziku)
TASTATURA	55 alfanumeričkih i funkcionalnih dirki
EKRAN	24 linije × 40 znakova
REZOLUCIJA SLIKE	240 × 216 tačaka
SET ZNAKOVA	64 (ASCII)
GRAFIKA	64 programirljivih znakova od strane korisnika (višebojna grafika)
BOJA	8 osnovnih boja, znakova i simbola
TON	8 oktava (1024 tonova) 16 nivoa jačine zvuka i specijalni efekti.
VIDEO MODULATOR	PAL sistem
SPOLJNI PRIKLJUČCI	koaksijalni konektor za antenski ulaz TV prijemnika koaksijalni priključak za ulaz u monitor. konektor za priključenje kasetovona konektor sistemske magistrale
SERIJSKI INTERFEJS	RS 232 C
MOGUĆNOST PRIKLJUČENJA PERIFERIJA	serijski štampač Mini-flopi disk
PROŠIRENJE ROM/RAM	16 KB za editor i asembler /16 KB Korisnički memorijski prostor.

ELEKTRONSKA INDUSTRIJA — NIŠ, **RO „EI-RAČUNARI", OOUR „FABRIKA RAČUNSKIH MAŠINA"** 18000 NIŠ, Bul. Veljka Vlahovića 80—8
plasman 018/54-779, 51-568, TLX. 16283
PREDSTAVNIŠTVA: 11000 BEOGRAD, Ul. Rudo 2, 011/488-260 81000 TITOGRAD, Ul. Brace Bracanovića 58, 081/34-739

računari 3
u vašoj kući

Specijalno izdanje časopisa „Galaksija"
Decembar 1984. | Cena 200 D

sadržaj

4/ vesti

8/ račun(aljka) u našim glavama

10/ računari u izlogu: amstrad, elektron plus, spektrum plus

15/ disk jedinice

21/ računar je muškog roda

22/ program od milion dolara

24/ neki bolji bejzici

29/ halo, da li je to mašinac?

32/ šta ejkorn nije rekao

34/ spektrum sa deset ruku

51/ nove naredbe na „komodoru"

56/ pedeset veličanstvenih

62/ ćirilica na epsonu

64/ ukleti dvorac i druge bajke

73/ letači bez diplome

78/ programator eproma

80/ ubiću se, neće da radi!

81/ da li si završio galaksiju

Poseban prilog: ROM 2 ZA „GALAKSIJU"

Izdaje
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga"
11000 Beograd
Bulevar vojvode Mišića 17

Telefoni
650-161 (redakcija)
650-528 (prodaja)
651-793 (propaganda)

Generalni direktor
Gojko Zečar

Direktor OOUR „Duga"
Zoran Milošević

Glavni i odgovorni urednik
Gavrilo Vučković

Urednik izdanja
Jova Regasek

Likovna i grafička oprema
Dušan Mijatović

Redakcija časopisa „Galaksija"
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika
Esad Jakupović, zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Srđan Stojančev, novinar
Gavrilo Vučković, glavni i odgovorni urednik

Stručna saradnja
Dejan Ristanović
Nevenka Spalević
Mihajlo Tešević, dipl. ing.
Anđelko Zgorelec

Autori tekstova
Predrag Bogdanović
Milan Ivanović
Miodrag Klajić, dipl. ing.
Vladimir Kostić
Srboljub Kuzmanović, dipl. ing.
Zoran Modli
Jova Regasek
Dejan Ristanović
Jelena Rupnik
Jovan Skuljan
Nevenka Spalević

Prevodioci
Esad Jakupović
Ksenija Pješčić—Lebedinski

Tehnička saradnja
Ljubiša Milovanović
Elizabeta Novak

Izdavački savet „Galaksije"
Dr Rudi Debijađi, prof. dru Branislav Dimitrijević (predsednik), Radovan Draškić, Tanasije Gavranović, Živorad Glišić, Esad Jakupović, Velizar Maslać, Nikola Pajić, Željko Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vesović, Miloje Vuković

Štampa
Beogradski izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17

Žiro-račun kod SDK
60802-833-2463
Devizni račun kod Beobanke
60811-620-6-82701-999-01066
Za inostranstvo cena dvostruka (400D, 2,50$, 6,50DM, 45Sch, 5,50Sfrs, 20Ffrs)
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika" broj 26/72, ovo izdanje oslobođeno je poreza na promet.

...ta ima novo

Nepismeno o pismenosti

Nedavno je u prostorijama jedne ozbiljne društvene organizacije u Beogradu održana sednica povodom Manifesta za računarsku pismenost koji je sredinom septembra izdala GK SSO. Po svom dobrom, starom običaju, članovi GK SSO, inače inicijatori ovog sastanka, malčice su zakasnili, ostavljajući prisutne u slatkom uverenju da sednice neće ni biti. I dok su se neki potajno, a neki i glasno nadali tome, drugi su pokušavali da reše zagonetku: šta to omladina omladina hoće da postigne ovim Manifestom? Složili su se i jedni i drugi: mladi opet ne znaju šta hoće.

Posle nepun čas pojavljaju se i pripadnici mladih snaga. Sednica napokon može da počne. Pričalo se mnogo o uvođenju računara u škole, o oslobađanju uvoza i prodaji preko konsignacije. Razgovor koji je ubrzo zatim „spontano potekao", pokazao je, međutim, da neke od prisutnih ti „računarčići" neodoljivo podsećaju na flipere (a ko je još video flipere u školama, pa oni nisu didaktički opravdani?!). Ali, neko njihovo „šesto čulo" govorilo im je da javno ne smeju biti protiv osnivanja Društva mladih informatičara, protiv opremanja škola i fakulteta računarima, jer je omladina, ovaj put, slučajno u pravu.

I kao što to na kraju ovakvih diskusija obično biva, jedini zaključak koji je predsedavajući mogao da donese jeste da su svi voljni da pomognu i pruže podršku omladini koja je, na kraju, ipak, znala šta hoće. Imamo Manifest, imamo reforme, zahvate, analize, programe . . . šta li nam ono još treba za uspešnu borbu protiv kompjuterske nepismenosti?

J. Rupnik

Prvenac iz Niša

Popularnost koju kućni računari stiču u našim podnebljima sve više podstiče naše industrijske gigante da se uključe u njihovu proizvodnju i osvoje deo gladnoga tržišta. Posle Iskre, koja je svakako jedan od pionira u ovoj oblasti, na nedavno završenom Sajmu knjiga je promovisan kućni kompjuter Elektronska industrija iz Niša koji će se,

kako se nadamo, naći u prodaji posle Nove godine.

Novi kompjuter je zasnovan na relativno retko korišćenom Cosmacovom mikroprocesoru 1802 (ako razgovarate sa nekim iz El Niša, ne pominjite ovu oznaku — procesor se zvanično zove samo CDP 1802). Snabdeven je više nego solidnim RAM-om od 32 K koji je praktično sasvim slobodan — video memorija zauzima odvojena 4K. Ova 4K su iskorišćena za pamćenje teksta, pri čemu u red staje po 40 slova, od kojih svako može da bude obojeno jednom od 16 boja ili, da budemo precizniji, jednim od dva intenziteta jedne od 8 boja. Grafika praktično ne postoji — može da se postigne jedino predefinisanjem karaktera. Oni koji prate kretanja na tržištu svakako se sećaju da je ovakva koncepcija grafike „usrećila" mnoge vlasnike danas nepostojećeg TI99/4A.

ROM od 16K sadrži RCA Bejzik 3. Radi se o veoma dobrom dijalektu standardnog bejzika sa značajnim matematičkim mogućnostima, ali bez procedura, lokalnih promenljivih i sličnih „novotarija". U neposrednoj budućnosti na tržištu će se pojaviti i RCA paskal.

Na Sajmu je obećano da novi kompjuter neće koštati više od 6 starih miliona, što bi za naše uslove bila izvanredna cena. Ima li, međutim, novi računar Elektronske industrije šansi na komercijalni uspeh pored „spektruma", „Komodora" i „akorn elektrona" koji će, kako izgleda, uskoro moći da se uvoze uz 46% carine? Umesto odgovora, reći ćemo da se El Niš oštro protivi legalizaciji uvoza. Govori li nam to nešto?

D. Ristanović

Meka za izdavače

Ovogodišnji Sajam knjiga u Beogradu ulio je malo nade ljubiteljima računara. Pojedine knjige iz ove oblasti već su izašle iz štampe (i već su rasprodate), za neke se pretpostavlja da će izaći do kraja godine, a pravi „bum" računarske literature kod nas trebalo bi očekivati tek iduće godine.

Interesovanje posetilaca za literaturu iz oblasti računarstva bilo je ogromno, što potvrđuju i podaci o rasprodatim knjigama. „Kompjuter u kući", knjiga koja je u izdanju „Cankarjeve založbe" štampana u tiražu od 8000 primeraka, već je rasprodata. Čini se da je „Cankarjeva založba" na vreme shvatila za čim domaće tržište već dugo čezne i na vreme odštampala „jednu od najtraženijih engleskih knjiga o kompjuterima".

Posle slovenačkog, iz štampe će izaći i srpskohrvatsko izdanje, i to u tiražu od 12000 primeraka. Ovo je jedna od prvih knjiga koja jasnim, netehničkim rečnikom, brojnim ilustracijama i šematskim prikazima uvodi čitaoca u svet računara. Knjiga je istovremeno vodič i priručnik od prvih koraka, „od trenutka kada se odlučite da nabavite računar do zapanjujućih novih granica u grafičkom dizajnu, robotici i veštačkoj inteligenciji". Cena iznosi 3300 dinara, a pouzećem se dobija 20% popusta.

Na štandu Cankarjeve založbe pažnju je privlačila i knjiga „BASIC za početnike". Ova knjižica od nekih 120 strana predstavlja vodič za nove korisnike ZX Spectruma, ali u njoj ima korisnih stvari i za dobre poznavaoce ovog računara, koji se odnedavno može naći i u našim prodavnicama. Drugovi iz „Cankarjeve založbe" verovatno znaju da se oni pravi zaluđenici za računare ne mogu prevariti sjajnim koricama i luksuznom opremom; „BASIC za početnike" je knjiga džepnog formata, štampana na papiru slabog kvaliteta, pa joj je i cena pristupačnija (400 din.).

„Mladinska knjiga" pojavila se na beogradskom sajmu sa još nedoštampanom

ŽUTO SVETLO ZA RAČUNARE

Zbog čega je jedna skorašnja odluka Saveznog izvršnog veća prvo obradovala, a onda u izvesnoj meri razočarala ljubitelje kućnih računara?

SIV je, kao što je poznato, dozvolio građanima („fizičkim licima") da mogu uneti „jedan lični (kućni) računar u vrednosti do 40.000 dinara, prilikom prvog dolaska iz inostranstva u toku godine". Praveći ovaj srazmerno značajan izuzetak u odnosu na ostale vrste roba — povoljniji tretman, praktično, imaju samo lekovi, te knjige i časopisi s područja nauke, umetnosti i kulture, koji mogu da se uvoze neograničeno i na koje se ne plaća carina — SIV je pokazao veliko razumevanje za mnogobrojne zahteve javnosti da se sa kućnog računara skine etiketa „luksuzne igračke" i da on počne da se smatra tehničkim pomagalom koje nas obrazuje i uvodi u tehnološku eru i bez koga već u bliskoj budućnosti život gotovo neće moći da se zamisli. Savezno izvršno veće je, dakle, pokazalo dobru volju prema svima koji se na bilo koji način bave kućnim računarima i za taj „sluh" javnost mu je već odala priznanje.

Na žalost, ta dobra volja je u izvesnom raskoraku sa samom formulacijom propisa, koji je, očigledno, previše pojednostavljen i manjkav. Praktično gledano, za iznos od 40.000 dinara u inostranstvu može da se kupi svega nekoliko računara: ZX 81, „spektrum" — a ako se propis malo nategne (za šta će carinici sigurno imati razumevanja) i „komodor 64". Za sve ostale računare, zabrana uvoza je i dalje na snazi, premda je reč o kućnim kompjuterima srazmerno većih mogućnosti, pa, znači, i većeg pedagoškog i praktičnog značaja.

Stvari stoje još lošije ako imamo uvid da za tu relativno skromnu sumu nije moguće nabaviti i solidnu periferijsku opremu, a da se bez nje moć računara višestruko smanjuje. Da budemo precizni, uz računar se može uneti i periferijska oprema, ali sve zajedno do sume od 40.000 dinara. To, praktično, znači da uz „spektrum" jedva možete da unesete njegov mali printer, a da uz „komodor" ne možete uneti ništa. Naredne godine moći ćete ponovo da unesete neki od pomenutih računara (što teško da će vam biti potrebno), ali ne i samu periferijsku opremu do iznosa od 40.000 dinara (premda će vam ona biti neophodna).

Mislimo, prema tome, da je propis o uvozu kućnih računara morao da bude rastegljiviji kada je reč o novčanoj granici, precizniji u pogledu vrsta kompjutera i motiva nabavke, predusretljiviji u smislu unošenja periferijske opreme kada je građanin već vlasnik računara . . .

SIV je, dakle, upalio svetlo za legalno unošenje kućnih kompjutera, ali zasad još uvek žuto. Korak je, međutim, ohrabrujući: možda ćemo nagodinu ipak dobiti — zeleno!

E. Jakupović

knjigom „Hišni računalnik" čija cena iznosi 3300 dinara, ali se prednarudžbenicom može nabaviti za „samo" 2800 dinara. Ovo je još jedno od luksuznih izdanja kojima domaći izdavači pokušavaju da nas zasene; prosta računica dovodi vas do saznanja da se popustom svaku stranicu ove knjige plaćate 12,5 dinara.

„Prosvjeta" iz Zagreba najavila je komplet engleskih autora pod nazivom „Svijet kućnih računala", koji treba da izađe iz štampe početkom naredne godine. Prva u nizu, knjiga pod naslovom „Sve o kućnim računalima" obrađuje osnovne principe rada svakog računara, a u njoj su detaljno opisani i svi delovi kućnih kompjuterskih sistema, od centralnog procesora do perifernih uređaja. Knjiga „Prvi koraci u BASIC-u" opisuje i objašnjava kako se pomoću tog programskog jezika rešavaju neki česti svakodnevni problemi, bez obzira na model računara koji imate. Uz mnoštvo primera, u knjizi se nalaze i zadaci koji će vas naterati na razmišljanje. Ukoliko tome niste vični, zgodnu okolnost predstavlja i to što se u knjizi mogu naći gotova rešenja. U trećoj knjizi, „Igre, grafika i zvukovi", pažnja je posvećena zabavi sa kućnim računarima. S obzirom na to da u tome glavnu ulogu imaju crteži i zvuk, autori objašnjavaju kako se u bejziku pišu programi koji koriste te osobine računara. Knjiga „Učenje uz računalo" opisuje kako se kućni računari koriste u sticaju novih znanja, kako pomažu u savladavanju školskih zadataka i sl. Ove knjige mogu se nabaviti samo u kompletu čija cena iznosi 4000 dinara.

U pripremi je još nekoliko naslova iz ove oblasti. Očekuje se da će u toku iduće godine izaći knjiga „Spectrum i programiranje". Drugovi iz „Tehničke knjige" nisu bili raspoloženi za priču, a cena ovog priručnika još uvek se ne zna, pošto je „nemoguće predvideti kako će se kretati cena papira". Po svemu sudeći, ova tajanstvena knjiga stići će u naše knjižare sa malim zakašnjenjem od nekoliko meseci do pola godine.

Nakon svega, ostaje utisak da se dovitljivi trgovci služe sitnim lukavstvima ne bi li što dublje zavukli ruku u džep nesretnim ljubiteljima dobre računarske literature. Prosto vas izluđuju luksuznim izdanjima, debelim koricama (onda vam knjiga izgleda veća i ogromnim proredima, a sve je to moglo da prođe kud i kamo skromnije, i — jeftinije.

J. Rupnik

Drugi o nama

Izreka da se ono što štampa nije zabeležila nije ni dogodilo — samo je pošalica. Ali je, s druge strane, ipak tačno da štampa služi kao pouzdano svedočanstvo da se nešto događa. To, razume se, važi i za svet računara.

Jugoslovenske mogućnosti na svetskoj kompjuterskoj sceni prilično su skromne, pod uslovom da se za njih izvan zemlje uopšte sazna. Da se kod nas na tom polju nešto događa pokazuje i činjenica da je Jugoslavija počela da se pojavljuje u evropskoj kompjuterskoj štampi. Zasad se to čini sa izvesnom dozom humora, i to na račun kontroverznog „Iskrinog" poteza sa preuzimanjem tehnologije za „spektrum" od 16 kilobajta.

No, ne treba biti sitničav. Biće, naime, vremena i za jedan ozbiljniji odnos prema nama.

Iz poslednjeg boja engleskog mesečnika „Sinclair User" prenosimo jedan kraći prilog i jedan mali strip.

Drug Klajv

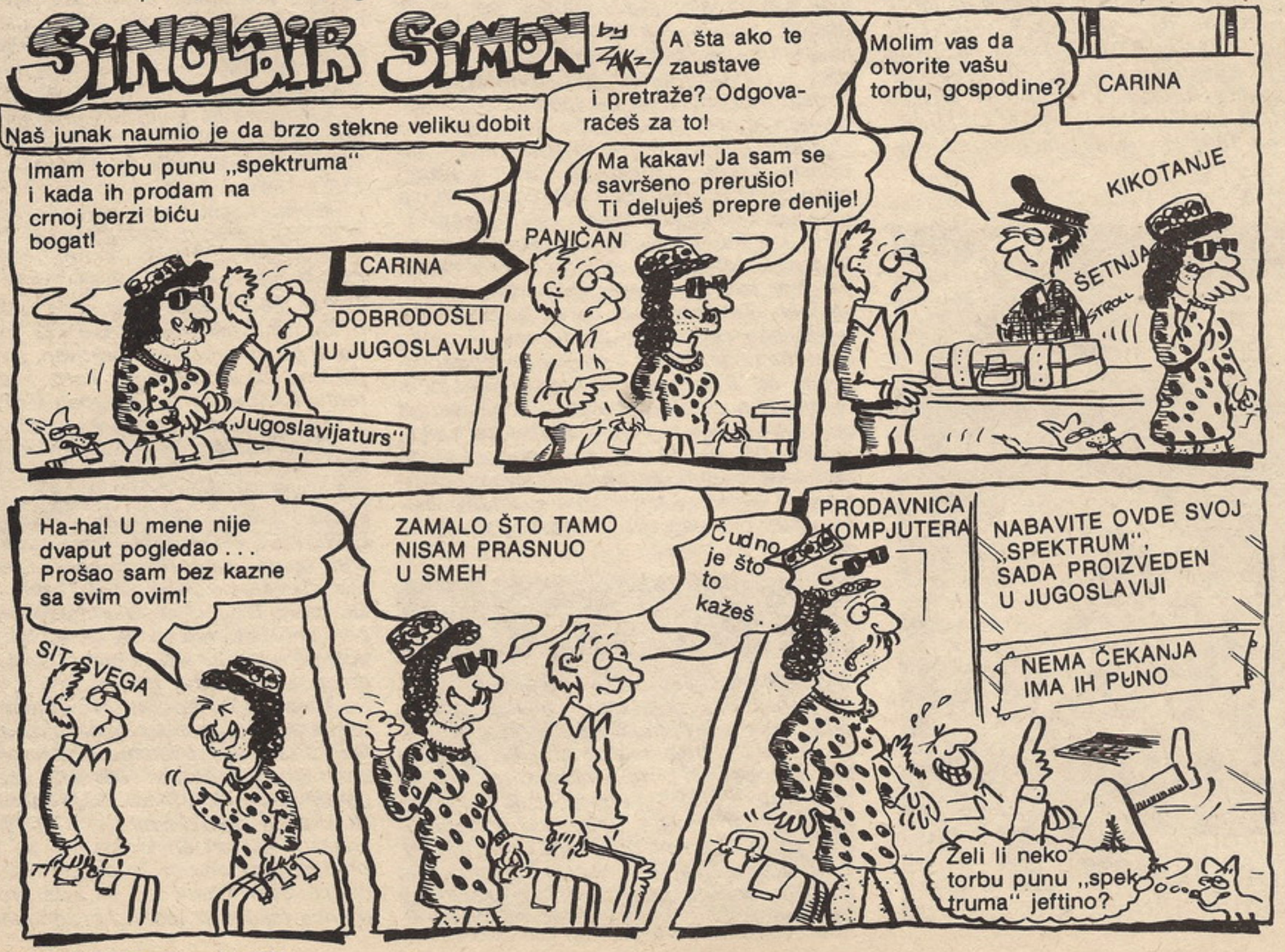
Švercovanje računara postalo je u Jugoslaviji veliki posao, pošto vlada s neodobravanjem gleda na uvoz mašina od strane pojedinaca.

Uprkos neslužbenoj zabrani, u vezi s mikrokompjuterima, male grupe bile su sposobne da mašine krijumčare u zemlju. Rezultat toga je da prošvercovano najmanje 20.000 mašina, pokrenuta su dva kompjuterska časopisa, a upravo je puštena i igra za „spektrum" o švercu kompjutera nazvana „Krijumčarena roba".

Sve to, međutim, počinje da se menja, pošto je „Sinclair Research" postigao dogovor sa fabrikom za elektroniku ISKRA za uvoz delova za „spektrum" u Jugoslaviju, gde će ove mašine biti sklapane i prodavane.

upitan da li je odluka doneta na nivou vlade, Džulijan Goldsmit (Julian Goldsmith) iz „Sinclair Research" kaže: „Naša prekomorska politika uvek je bila da radimo sa britanskim ambasadama i konzulatima, pa mislim da je posredi takav slučaj".

Jugoslovenska ambasada nije mogla da rasvetli stvar i uputila nas je da stupimo u kontakt sa Privrednom komorom Jugoslavije. Kao što kaže zvanični predstavnik: „Ne postoji zabrana uvoza kompjutera, ali osoba koja nastoji da unese računar u zemlju treba da plati izvesnu carinu. Bojim se da ne mogu ništa više da kažem, pošto se to tako često menja".



Prva YU kaseta

Otvaramo prvu domaću kasetu za 2X spectrum 48K. Prvi utisak je vrlo dobar. Kaseta je grafički lepo opremljena. Unutra je i mala knjižica sa uputstvima od 24 strane. Kasetu je izdao „RADIO STUDENT" iz Ljubljane. Srpskohrvatska verzija (koju vam predstavljamo) pojavila se povodom nedavno zatvorenog Sajma knjiga u Beogradu.

Na samoj kaseti se nalazi čitavih deset programa! Svi su prevedeni na srpskohrvatski — sem jednog, najboljeg. Radi se o avanturi „KONTRABANT". Cilj igre je više nego aktuelan: prokrijumčariti kompjuter! Kroz igru defiluju Martin Krpan, Primož Trubar (priznati šverceri), Pate Marelov, veštice i drugi.

Ostali programi su pretežno uslužne namene. Tu je analizator zvuka, program za crtanje, traser, englesko-slovenački rečnik (!) od preko tri hiljade reči, igra podmornice i još nekoliko.

Šta na kraju reći? Većina programa, da budemo iskreni, nije vrhunskog kvaliteta, ali je od toga mnogo važnije da se neko osmelio da se osloni na domaću programersku pamet. Napraviti pravi profesionalni program zahteva višemesečni rad grupe vrhunskih programera. Sve dok armija preprodavaca bude bespravno presnimala programe, takav posao se kod nas neće isplatiti.

Kolegama iz RADIO STUDENTA-a želimo puno uspeha u daljem radu. Onisu već najavili nastavak avanture „KONTRABANT" za kraj godine. Ujedno pozivaju sve koji su sami napisali neki dobar program da im se jave radi saradnje. (RADIO STUDENT/61 000 LJUBLJANA) STUDENTSKO NASELJE BLOK 8). Kasetu možete naručiti na adresu: MLADINSKA KNJIGA, KNJIGARNA TITOVA 3, 61000 LJUBLJANA, Tel: 061/211-895. Cena od 1300 dinara se plaća pouzećem

V. Kostić

Inteligentni telefon

Engleska firma „STC Telecommunications" pustila je u prodaju „Executel", inteligentni telefon koji zadovoljava gotovo sve potrebe jednog šefa poslovnice ili direktora preduzeća. Telefon kombinuje mogućnost automatskog biranja ili biranja preko dugmadi, razgovor preko slušalice ili zvučnika (tako da su ruke slobodne), sa teleks terminalom, kalendarom rada za 20 godina, ličnim adresarom i telefonskim imenikom, beležnicom, računarom, časovnikom i zvučnim alarmom koji treba da podseti korisnika na predstojeći sastanak. Telefon može, takođe, da se poveže sa službom „Viewdata" i šalje i prima elektronsku poštu. Ukoliko je nekom i pored ovog telefona potrebna sekretarica, firma mu nudi i specijalni sekretarski priključak.

U kućištu niskog profila smešteni su tastatura standardne veličine i obrtni ekran, što čini telefon „Executel" i vizuelno privlačnim. Telefon može da se koristi preko telefonske centrale u preduzeću ili preko direktne linije. „Srce" uređaja je mikroprocesor od 8 bita sa jednom integralnom memorijom od 64 kilobajta i minikasetom od 32 kilobajta za lične podatke korisnika. Informacija je od neovlašćenog pristupa zaštićena šifrom. Zahvaljujući svom savremenom izgledu i tehničkom kvalitetu, telefon je osvojio nagradu Britanskog saveta za dizajn.

Biju ko neće

Posle čekanja od skoro godinu dana, QL se, konačno, pojavio u londonskim prodavnicama.sa najavljenom cenom od 399 funti. Za razliku od prve (nedovršene) verzije koja se i isporučivala pretplatnicima, ceo operativni sistem QDOS i superbejzik stali su u ROM unutar kućišta. Uz računer se isporučuju četiri poslovna programa firme PSION, četiri prazna mikrodrajv kartridža, ispravljač i profesionalno opremljeno uputstvo za upotrebu.

Od periferija za sada mogu da se kupe RGB monitor (nažalost, izlaz za kompozitni video monitor nije predviđen) koji proizvode više firmi sa oznakom QL po cenama 200—300 funti i printer — svaki koji ima RS 232C serijski interfejs (sa cenama od 200 funti naviše). Neke nezavisne firme su prilagodile svoje disk jedinice i hard-diskove za priključenje na QL, ali su cene neprimerene računaru ove klase. S druge strane, Sinkler je već objavio da uskoro, po niskim cenama, iznosi na tržište nekoliko proširenja: RAM karticu kapaciteta 500 K, interfejse za paralelnu vezu, za „vinčester" disk i IEEE-488, zatim modem i a/d konverter.

QL se nalazi izložen u skoro svakoj prodavnici koja prodaje kompjutere — interesenata je mnogo a kupaca malo. Kupci čekaju novi softver i hardverska proširenja, a firme koje bi to trebalo da razviju više kupaca QL-a i krug se tu — zatvara. Ostaje utisak da je QL još uvek nedovršen, ako se na njega gleda kao na poslovni mikrokompjuterski sistem. Na njegovom dovršenju za sada punom parom radi samo Sinkler, pa od njegovih rezultata najviše i zavisi dalja sudbina QL-a na tržištu.

M. Klajić

Prvi programi za QL

Jedan od glavnih razloga slabe prodaje QL-a je, bez sumnje, nedostatak softvera. Osim četiri poslovna programa, koji se dobijaju uz računar, u prodavnicama se ne može kupiti ništa više. Sinkler uskoro najavljuje C kompajler, 68000 asembler i terminal emulator. Za sada se mogu naručiti samo programi nezavisnih proizvođača.

QUEST International Computers objavio je dva paketa knjigovodstvenih programa sa standardnim knjigama prodaje, faktura, kontrolu zaliha itd. Paketi se zovu Tally 1 i Tally 2; pisani su za QDOS i prodaju se za 115, odnosno 57 funti.

Možda najznačajniji softverski paket koji je do sada predstavljen za QL dolazi takođe iz laboratorija Questa — operativni sistem CP/M 68K, vrlo blizak sa CP/M 2.1. On sadrži 68000 asembler, sve standardne CP/M sistemske komande i C kompajler. Da bi sistem postao operativan, potrebna je još tzv. OS ekspanziona karta kao i proširenje RAM-a za 64 K (postojeći RAM ne može istovremeno da opslužuje CP/M i QDOS). Najmanji softversko/hardverski paket za podršku CP/M-a košta 178 funti. CP/M koristi sve QDOS sistemske pozive, pa sadrži i većinu QL-ovih bagova (napr CTRL ALT 7). Može da radi sa mikrodrajvom (kada je neprijatno spor) ili sa disketom. Questo će preko CP/M-a predstaviti za QL puni izbor svog Padmede poslovnog softvera. Očekuje se da će većina Questovog softvera za QL naći na rafovima prodavnica uoči Nove godine.

Alternativni operativni sistem za QL predstavila je takođe i malo poznata firma GST Computer Systems iz Kembridža. Sistem pod oznakom 68K/OS razvijen je paralelno sa QDOS-om, ali je zahtevao više ROM-a od predviđenih 32 K. To je, inače, snažan višenamenski sistem sa mnogim prednostima preuzetim od „Uniksa", uključujući istovremeni rad sa više programa i prenos fajlova iz jednog programa u drugi

kroz „filter" programe za modifikaciju podataka. Sistem 68K/OS (100 funti) dobija se na pločici koja se ubacije u unutrašnji ekspanzioni port za proširenja. Asembler je 40 funti. Glavni nedostatak ovog operativnog sistema je u tome što za njega za sada ne postoji nikakav aplikativan softver. Njegov uspeh na tržištu zavisi od toga da li će ga prihvatiti ostali proizvođači softvera. Možemo, međutim, da zažalimo što Sinkler nije dodao još neki ROM čip i ovaj OS preuzeo kao glavni za QL.

M. Klajić

Ejkorn u poslovnoj klasi

Posle velikog uspeha računara BBC i electron, firme Ejkorn je, očigledno, rešila da se iskuša u proizvodnji i personalnih računara najviše klase. Rezultat je ABC serija kompjutera (ABC je, doduše, skraćenica od Acorn Bussiness Computers ali, priznaćete, mnogo podseća na BBC!) koja je najavljena i (delimično) prikazana na nedavnom sajmu Acorn Usera.

Osnovni model ABC računara je jednostavno BBC B sa numeričkom tastaturom smešten u novoj kutiji. Računaru je pridodat monitor i dve disk jedinice po 760 Kb — Ejkorn je, konačno, usvojio dvostruku gustinu upisa.

Sledeća verzija je vrlo slična računaru BBC B sa pridodatim procesorom Z80A i, u zbiru, 96 Kb RAM-a. Posetioci sajma su imali prilike da vide i prave novitete, proširenja ABC modela Intelovim super procesorom 80286 i, sa softverske strane, implementaciju čuvenog programerskog operativnog sistema „Unix". U budućnosti će se, za specijalne primene, proizvoditi ABC kompjuteri sa Intelovim mikroprocesorom 80386 koji bi trebalo da imaju snagu i brzinu velikih Digitalovih računara.

Ejkorn se pokazao vrlo poslovno obećavajući da će vlasnici BBC B moći da se prošire do bilo koje od ovih konfiguracija uz „prihvatljiva ulaganja". Svi vlasnici BBC računara bi vrlo rado saznali kolika su ta prihvatljiva ulaganja, ali se Ejkorn ne izjašnjava — cene će biti objavljene tek kad se kompjuteri, pred Novu godinu pojave u radnjama.

D. Ristanović

Ejkorn otkupio Torč

Firma Torch Systems Ltd. se poslednjih godina istakla proširenjima BBC B koja su bila jeftinija i ažurnija od Ejkornovih. Ejkornu ova konkurencija nije mnogo smetala (što više hardvera za njihov računar, to bolje), ali su direktori ove firme svakako strahovali od nekog nepredvidljivog poteza Torča. Rezultat — Ejkorn je otkupio dovoljno akcija Torča da ovu firmu drži pod kontrolom, ali joj je ostavio maksimalnu samostalnost.

Torč i dalje prodaje izvanredne pakete koji se, u maksimalnoj konfiguraciji, sastoje od masivnog diska, jedne disk jedinice, dodatnog procesora Motorola 68000 i procesora Z80A. Ovu konfiguraciju kontrolišu operativni sistemi CP/M „Unix". Cena proširenja nije prevelika (oko 2000 funti) za ono što nudi, ali je softver strahovito skup.

Posle ove kupovine, Ejkornove akcije su nešto porasle. Iako je u toku 1983. Ejkorn prodao daleko više kompjutera nego što su očekivali najveći optimisti, njegove akcije su bile u stagnaciji i padu što se tumači ogromnim ulaganjima u prodor na američko tržište, koja i dalje daju vrlo slabe rezultate.

D. Ristanović

Dva nova „Komodora"

Januara ove godine u Las Vegasu su prvi put javnosti prikazana dva nova Komodorova računara, bazirana na novom mikroprocesoru 7501. Softver za ove računare nije kompatibilan sa softverom za bilo koji već postojeći Komodorov računar, ali u firmi kažu da su novi modeli napravljeni da popune izbor računara, a ne da zamene već postojeće.

Zovu se „plus 4" i V364 (mada imaju i druga imena kao, na primer, 64+) i izgledaju sasvim drugačije od Komodorovih mašina koje se mogu naći. Tastatura i kutija su drugačije i po spoljašnjem izgledu „plus 4" i V364 podsećaju na japanske računare.

Kao što samo ime govori, oba računara imaju 64K RAM korisničke memorije. „Plus 4" ima ukupno 32K ROM i 64K RAM-a od kojih je 60K adresivo iz bejzika. Grafičke mogućnosti se dosta razlikuju od mogućnosti C64: visoka rezolucija je i dalje 320×200 tačaka, međutim „264" ima 121 moguću boju, i, za razliku od C64, nema sprajtove. „Plus 4" ima i uprošćenu verziju audio čipa, sa dva kanala. Svaki od ovih kanala može generisati muziku ili šum.

Tastatura sadrži 67 tastera, od kojih su četiri programabilna. U kombinaciji sa „SHIFT", to daje osam funkcijskih tastera. Potpun set velikih/malih slova i grafičkih znakova nalazi se u ROM-u, koji takođe sadrži mašinski monitor, mogućnost deljenja ekrana, jak bejzik i dr.

Pored ovoga, sa računarom se mogu dobiti i ugrađeni ROM-ovi koji sadrže program za obradu listi, program za obradu teksta, bazu podataka i dr. Svi ovi programi, ukoliko se računar kupi bez njih, mogu se nabaviti na disku ili kartridžu.

Komodor V364 je vrlo sličan sa „Plus 4". On u sebi još ima sintetizor glasa sa rečnikom od preko 250 reči. Dodatne reči se mogu učitati sa diska, ili se mogu nalaziti u kartridžu. ROM je povećan na 48Kb. Pored svih tastera koje sadrži „264", tastatura od 86 tastera uključuje i numeričku tastaturu sa 19 tastera.

P. Bogdanović

Brže disk jedinice

Micro Systems Development inc. je izbacio na tržište drajvove MSD-SD1 i MSD-SD2 (SD — Super Disk), kompatibilne sa „komodorom 64". Oba drajva koriste diskete prečnika 5,25 inča i formata „2a" (isti format koriste i 1541, 2031, 4040...): 170Kb, 35 staza... Kao i sve Komodorove disk jedinice, i MSD drajvovi imaju svoj procesor, 16Kb ROM-a i 4Kb RAM memorije koja se koristi za bafere i potrebe operativnog sistema. DOS MSD-a baziran je na procesoru 65110 i potpuno je kompatibilan sa „komodorovim" DOS-om). Za povezivanje sa računarom koristi se serijski IEC bas preko koga se ostvaruje ista veza kao i između npr. CBM-64 i VIC-1541. Pored serijskog IEC basa, tu je i Paralelni IEEE-488 interfejs, tako da se ove disk jedinice mogu koristiti sa bilo kojim računarom koji je opremljen ovim interfejsom.

Unutrašnje operacije u ovim jedinicama su brže nego kod 1541. Formatiranje diskete, na primer, traje samo 17 sekundi, dok će MSD-SD2 kopirati celu disketu pomoću duplicate komande za manje od 50 s! Međutim, prenos podataka kroz serijski bas je i dalje spor kao i kod 1541. Za brži prenos može se koristiti IEEE-488.

Sve u svemu, MSD disk jedinice predstavljaju dobru zamenu za Komodorove (po svemu sem po ceni) s obzirom da se mogu koristiti uz bilo koji Komodorov računar bez ikakvih interfejsa. Naročito je značajna pogodnost u radu koju pružaju sve dvostruke disk jedinice kao i MSD-SD2.

Jedan od brzih diskova za CMB-64 je QUIKDISK firme MICROTECH. On se direktno priključuje na CMB-64 preko porta za proširenja. Procesor računara kontroliše disk tako da se svi podaci direktno učitavaju u mermoriju. Uz standardnu disk jednicu dobija se i kartica koja sadrži DOS.

Operativni sistem je P-DOS koji emulira Komodorov drajv. Tako se i ovoj jedninici dodeljuje fizički broj 8, koji se može promeniti. Komande za rad se sekvencijalnim tekama, formatizaciju i inicijalizaciju diskete, kao i učitavanje i snimanje programa, iste su kao i za VIC-1541 ili bilo koji drugi Komodorov disk. Sve ostale operacije sa diskom rade se preko posebnog programa.

Brzina čitanja, u ovom slučaju i brzina prenosa, jer se svi podaci smeštaju direktno u memoriju, iznosi 250000 bauda. To je više od 15 puta brže nego brzina prenosa kod 1541.

Na tržištu se može naći više modela ovih disk jedinica:

340-2 kapacitet 286Kb, 3 inča, dvostruka gustine zapisa.

580-1 kapacitet 286Kb, 5,25 inča, jednostruka gustina zapisa.

580-2 kapacitet 572Kb, 5,25 inča, dvostruka gustina zapisa.

P. Bogdanović

račun(aljka) u našim glavama

Mikroračunarska revolucija na jugoslovenski način

O graničenjima koja se manifestiraju u našem društvu — kada je riječ o kućnim računarima — govorilo se i pisalo u javnosti tokom posljednjih godinu dana s manje ili više gorčine, ali većinom načelno, uz uopćene ilustracije iz inozemne prakse. Malo je bilo pokušaja da se sitemski pristupi pitanju kompjuterizacije kućnim računarima. Istina — sistemski pristup zahtijeva odgovovarajuću inofrmatičku podlogu, a u našim uvjetima, sa bezbroj ograničenja i nepoznanica, teško je doći do ozbiljnijih podataka. Tako će i ovaj napis biti samo nastojanje da se izvrši okvirna analiza tržišta programa za kućne računare i na taj način pridonese približem definiranju obima stupnja razvijenosti kompjuterizacije kućnim računarima u našoj zemlji.

Prije iznošenja određenih pretpostavki i zaključaka, potrebno je, makar ukratko, naznačiti najvažnije činioce okruženja i elemenata u društvu koji su umnogome utjecali da tržište programa za kućne računare bude baš takvo kakvo je danas.

Računarija kao elementarna nepogoda

Polazna hipoteza u ovom napisu može se definirati ocjenom da nas je talas kućnih računara iz razvijenih zemalja pogodio kao elementarna nepogoda — nenajvaljeno i neočekivano. Jer, kućna kompjuterizacija (*pod tim se podrazumijeva i ulazak malih računara u škole, u manja poduzeća, i u poslovanje zanatlija)* u nas nije logično nastavljanje procesa primjene kompjutera u privredi i društvenim službama — kao što je to bio slučaj u drugim zemljama. I čini se da je tu jedan od važnijih izvora naše ukupne društvene inertnosti. Daljnja — sasvim logična — posljedica ovog procesa je svojevrsna nelegalnost svega onoga što je vezano za kućne računare u nas (u ogromnom dijelu) uvažavajući zvanične carinske zakone — gotovo 90 posto kućnih računara je ilegalno uneseno u zemlju.

Još je nekoliko ograničenja razvoju kućne kompjuterizacije (u stvari, radi se o demokratizaciji informatike); najvažniji, čak važniji od carinskih ograničenja (jer se ne može preko noći izmijeniti usvajanje nekog novog zakona) vezan je za ekonomski status, za platežnu sposobnost prosječne porodice u našoj zemlji. **To se odnosi i na škole svih stupnjeva.** Za naše prilike cijena personalnog kompjutera je r e a l n o nekoliko puta veća od sličnog izdatka prosječne porodice u nekoj srednjerazvijenoj zemlji. Kućni računar nije dostupan prosječnom građaninu u našoj zemlji (prosječnom po imovinskom stanju i prihodima) kako je dostupan u drugim zemljama. To se odnosi i na škole svih stupnjeva. U evropskim zemljama kućni računar prosječnih perfomansi stoji oko 1/3 mjesečne nadnice prosječnog radnika, a u nas je to iznos od tri prosječna mjesečna osobna dohotka zaposlenog u organizaciji udruženog rada.

Daljnja prepreka našeg uključivanja u svjetsku informatičku revoluciju je veliki stupanj neobavještenosti i odsustva informacija o kvalitetama i mogućnostima pojedinih tipova računara u širim razmjerama. Ako ovom dodamo i činjenicu da postoji sasvim određena jezička prepreka do saznanja iz inostrane literature u širim razmjerama, kao i gotovo nikakve mogućnosti šire nabavke i distribucije časopisa i knjiga iz ove oblasti iz inozemstva (a naših nemamo) — onda je ograničenost potpunija. Da sve te neizvjesnosti budu kompletne pridonosi i nepovoljna situacija sa servisima za održavanje i popravak kućnih računara.

Ko to onda (i zašto) u ovoj zemlji, unatoč svih ovih nepovoljnosti, nabavlja kućni računar? Odatle i pitanje — u koje svrhe i kako se koriste kućni računari u nas. Odgovor u značajnom dijelu može dati — tržište.

Neligitimnost koja se nastavlja

Da je tržište programa za kućne računare potrebno ne treba previše dokaza. Primjena računara je toliko raznovrsna da praktično nije moguće tražnju zadovoljiti šablonskim paketom programa uz računar, a realna je pretpostavka da ne treba očekivati da će svaki vlasnik kućnog računara sam izrađivati sve programe koje će koristiti.

Tržište kućnih računara i programa kao i literature za ove računare u Jugoslaviji nije transparentno i ne postoji objektivni način da se utvrdi broj vlasnika računara, tipovi kompjutera koji ovi posjeduju, a još manje da se profiliraju interesne grupe i afiniteti za primjenu kućnih računara. Ispitivanje pulsa nezvaičnog tržišta programima za kompjutere kućnog formata u nas je moguće u okviru ograničenih mogućnosti koje pružaju mali oglasi u dnevnoj i tjednoj štampi i periodičnim izdanjima u našoj zemlji. Ovom prilikom analizira se stanje na hrvatsko-srpskom govornom području.

OGLAŠAVANJE — Oglašavanje tražnje i ponude programa za kućne računare (osnovni činilac tržišne komunikacije) prva je u nizu slabih karika u ovoj oblasti. Ne postoje redovna izdanja za korisnike kućnih računara (prvi su potezi tek eksperimentalno povučeni — a o profilu tih izdanja potrebno je diksutirati). Mali oglasi u dnevnim, tjednim i ostalim izdanjima nisu otvorili posebne rubrike za kućne računare — da ne govorimo o podjeli rubrika na hardver i softver ili na ponudu i tražnju — tako da se nezvanična tržišna komunikacija korisnika kućnih računara obavlja u okviru malooglasnih rubrika „prodaja-različito", „usluge" ili pod „ostalo". To se u potpunosti uklapa u jedno specifično stanje u nas koje se može nazvati „ne činiti atmosferu kompjuterizacije". Posebno je pitanje cijena oglasa koje su prevelike (izuzimajući „Galaksiju") i ne vide razliku između oglasa o prodaji polovnog (ili čak novog) „audi-a 100" i ponude za razmjenom ili prodajom programa za kućni računar. Popust od tridesetak posto na oglase o kućnim računarima neće baš tako ugroziti ekonomski položaj naših novinskih kuća.

Malo je tragikomično da neodgovarajući tokovi informacija (u štampi) ne omogućavaju razvoj kompjuterizacije kućnim računarima.

TRAŽNJA — Tražnja za programima kućnih računara zvanično i ne postoji. Gotovo da nije nađen oglas u malim oglasnicima u kojem se traži nekakav program za kućni računar. Zbunjeni sretnik koji je tek nabavio kućni računar i ne pomišlja oglasiti se u javnosti da mu je potrebna literatura, da su mu potrebne upute i programi za njegov računar. Tražnja se, tako, mora zadovoljiti postojećom ponudom — bez mogućeg utjecaja na asortiman, kvalitet i cijenu. O tome da bi neko oglasio tražnju za uslugom koje se može obaviti pomoću kućnog računara ne treba ni pomisliti.

A tražnja je značajan kreator, motiv i akcelerator razvoja neke proizvodnje ili obavljanja usluga — u našem slučaju: programa za kućne računare.

Rat cijenama do istrebljenja — kreativnosti

PONUDA — O ponudi se već može nešto više napisati — iako ne mnogo pohvalnije. Ponuda je, za sada, koncentrirana na privatni sektor (i to ne mora biti zlo), na pojedince — entuzijaste koji pomalo svoj entuzijazam zamjenjuju trgovačkim manirima (što već može biti zlo — ukoliko nema kvalitete).

Ponuda programa za kućne računare je disperzirana i nije privilegija samo glavnih gradova naših republika i pokrajina. Pored Beograda, Zagreba, Ljubljane i Skopja, i drugi veći gradovi imaju svoje entuzijaste koji nude „super software" (kako svi zaredom ističu); Niš, Karlovac, Osijek, Bor i desetak još manjih gradova zastupljeno je svojim entuzijastima u malim oglasnicima kada je u pitanju ponuda programa za kućne računare.

Na tom nezvaničnom i atomiziranom tržištu nude se uglavnom programi za kućne računare iz porodice **Sinclair** (ZX 81 i Spectrum) i Commodore 64, a poneki put i za naš domaći kućni računar „galaksija". U nastavku će biti analizirana ponuda programa samo za „Commodor 64", ali u osnovi ocjene stanja mogu se podrazumijevati i za računare ostalih tipova.

Kada se dobiju zatraženi katalozi po malom oglasniku (besplatno ili uz manju naknadu) može se zaključiti slijedeće:

Tržište programa za kućne računare veoma je značajan indikator razvijenosti i obima primjene kućnih računara. Na tržištu se susreću tražnja i ponuda određenih roba — u ovom slučaju programi i literatura za kućne računare — i njegovom se analizom dobija uvid u tražnju, ponudu i međuodnos ponude i tražnje kao najsnažniji barometar kretanja u ovoj oblasti. Što se na osnovu (ne)postojećeg domaćeg tržišta dade zaključiti o mikroračunarskoj revoluciji na jugoslavenski način?

— Šta su ta tvoja šezdeset i četiri kilobajta memorije u poređenju čak i sa najmanjom galaksijom, sa milionima njenih solarnih sistema i desetinama miliona sunaca?

Ponuda programa za kućne računare C-64 je neoriginalna, nemaštovita i plagijatorska (da se ne upotrijebi neki drugi izraz). Gotovo sve su ponude zasnovane na istom fondu programa koji se sastoji iz dvjestotinjak igara (sve su iz inozemstva), dvadesetak compilera i asemblera (sve iz inozemnih kompjuterskih laboratorija), desetak muzičkih brojeva (također iz inozemstva) i nekoliko desetaka programa tzv. korisničke naravi — od proračuna transformatora, bioritma, do učenja latinskih deklinacija (sve to opet iz inozemstva).

Većina entuzijasta nudi programe bez uputstava za primjenu — što je velika nevolja za one koji se odluče za kupovinu ove vrste zbog nižih cijena. Da bi bilo jasnije — radi se o prodaji programa za igre i ostale navedene svrhe koji su presnimljeni na kasetama ili u boljem slučaju na disketama. Svi ti programi su iz inozemstva, autorski su označeni i pravno zaštićeni tako da je i legalitet ovakvih preprodaja kopiranih programa po piratskom sistemu pod velikim znakom pitanja.

Cijene ponuđenih programa za kućne računare najbliji su indikator strukture, kvalitete i obima ponude. Proljetos, ove godine, cijene takvih snimljenih programa (nakupca kasetama) kretale su se od 700 do 1.300 dinara u većini slučajeva — da bi se svakom novom kupovinom broja prodavalaca povećavao geometrijskom progresijom: dojučerašnji kupac postaje prodavač u pola cijene (uz manje ili više truda oko kataloga koji nudi svoju kopiranu robu). Cijene su u novembru ove godine za iste programe pale na 80 dinara po igri ili drugom programu kod onih najnovijih prodavača (dojučerašnjih kupaca) tako da se već javljaju oglasi tipa „igre za Commodor 64 jeftino — samo da ne poklanjam". Ova vrsta suvremenog piratstva već je u začetku kompjuterizacije **obeshrabrila kreativnost** – time i mogućnost da se javljaju domaće softwerske grupe sa svojim programima.

Na što se onda svodi nabavka kućnih računara u nas (ovdje se izuzima primjena u edukativne svrhe u školama i klubovima tehnike — o kojoj se isto tako može napisati jedan oveći napis)?

Šansa koja se (ne) ponavlja

Na temelju uvida u strukturu i zastupljenost pojedinih grupa programa za kućne računare, njihove namjene, sadržaja i porijekla — a prema tridesetak kataloga sa nezvanično-zvaničnog tržišta na području koje smo analizirali — zaključujemo: personalni kompjuteri su skupa igračka u vrhu modnog trenda. Kompjuterske igre su ono što čini osnov tržišta.

Radi se o uklapanju Jugoslavije u tekuću informatičku revoluciju, o fascinantnoj tehnologiji obrade podataka što čini temelj suvremene privrede, obrazovanja i upravljanja. Društvo bi u cjelini (da ne pominjemo sada redom odgovarajuća ministarstva) moralo najozbiljnije razmotriti sadašnje stanje u oblasti kućnih računara i odgovarajućim mjerama učiniti da ne propadne šansa koja se ponavlja. Kod ovoga je potrebno razlikovanje proizvodnje kućnih računara i opreme — kao važnog područja tehnološkog nivoa zemlje — od važnog pitanja kompjuterske pismenosti nacije. Pri tome se ne smijemo zadovoljavati sa proizvodnjom skupih kućnih računara čija je proizvodnja u razvijenim zemljama napuštena.

Veliki novac koji su građani u zemlji sami investirali u kupovinu računara(sada govorimo o opismenjavanju nacije, a ne o kupovini roba u inozemstvu kao podrivanje vlastite industrije) — sada mora biti kanaliziran u nekoliko pravaca.

Jedan od osnovnih smjerova organizirane društvene akcije morao bi biti na opismenjavanju **mladih generacija** i njihovo pripremanje za društvo u kojem će stasati.

Drugi je pravac primjena računara za razne stručne poslove, znači za **generacije u naponu,** koje sada rade. Obrada podataka za razne ekonomske i druge analize, proračuni u mašinogradnji, građevinarstvu, elektronici i drugim privrednim djelatnostima, vođenje evidencija i knjigovodstva kako za kućni budžet, tako i za zanatlije i manje privredne organizacije, primjena ovih računara u stručno-istraživačkom radu pojedinaca ili cijelih timova . . . i t d. Primjena računara je isuviše kompleksna da bi se u nekoliko rečenica mogla detaljnije opisati. Njihovim uključenjem u privredni i društveni život omogućit će se postizanje daleko većih efekata u proizvodnji, i u isto vrijeme razvijat će se nove i nove djelatnosti, otvarat mogućnost novih zapošljavanja, dizat će se ukupna tehničko-tehnološka razina zemlje, mijenjat će se struktura društva.

Onaj treći pravac primjene kućnih računara u kojima se javljaju kompjuterske igre značajan je dijelom kao sredstvo suvremene i moderne zabave, i kao specifičan metod otklanjanja nesigurnosti u prvim susretima sa nepoznatim (kućnim računarom).

Stvaranje softverskih grupa koje bi mogle zadovoljiti tražnju za programima koje smo okvirno naznačili je proces koji zahtijeva vrijeme — ali isto tako i odgovarajuće uvjete. Mnogo se toga u našim propisima, shvatanjima i navikama mora izmijeniti — da bi uopće mogli početi sa nastojanjima da uhvatimo nekakav korak sa svijetom. Istina — i mnogi od nas takvo vrijeme računara će ostaviti iza sebe. Dosadašnja praksa — koju bi naša razna ministartva morala analizirati — nedvosmisleno je pokazala da smo u odnosu na Evropu još uvijek u vremenu računaljke. Ko zna — možda je to baš i nečiji račun. Račun koji ne poznaje računare već samo računaljke.

Milan Ivanović

amstrad cpc 464

Tržište kućnih računara još od 1980. priželjkuje novi veliki prodor sličan onome koji je učinio Klajv Sinkler svojom čuvenom ZX porodicom. To se, po mnogima, dogodilo upravo ovo jeseni i to sa strane sa koje se malo ko nadao — novi prodor je načinio Amstrad, firma koja se do sada praktično nije bavila računarima. Pomalo kriptično ime prvog Amstradovog kompjutera, CPC 464, dominira mnogim engleskim popularnim časopisima. Šta je CPC 464 učinilo toliko interesantnim I, kako saznajemo, komercijalnim?

Hardverska koncepcija CPC 464 je sasvim klasična: mikroprocesor Z80A, ULA čip koji kontroliše video, trokanalni generator tonova, standardni paralelni interfejs za štampač i port za džojstik. Amstrad je, ipak, uveo i nekoliko novina, od kojih su najznačajnije ugrađen specijalan kasetofon i crno beli monitor koji je uključen u osnovnu cenu od 200 funti.

Na prvi pogled, CPC 464 podseća na terminal u nekom računskom centru — dominira profesionalna „kolor" tastatura sa izdvojenim numeričkim delom, tipkama za pomeranje kurzora, kao i tasterima sa specijalnim funkcijama. Korisniku je ostavljena primamljiva mogućnost da dodeli proizvoljne stringove i funkcije mnogim tasterima i tako olakša kucanje komplikovanijih bejzik programa. Dobra tastatura i fina grafika koja omogućuje da se u jednom redu ekrana radi sa 80 slova čini „amstrad" najjeftinijim računarom na tržištu koji može izvanredno da se koristi za obradu teksta.

Ekran na BBC način

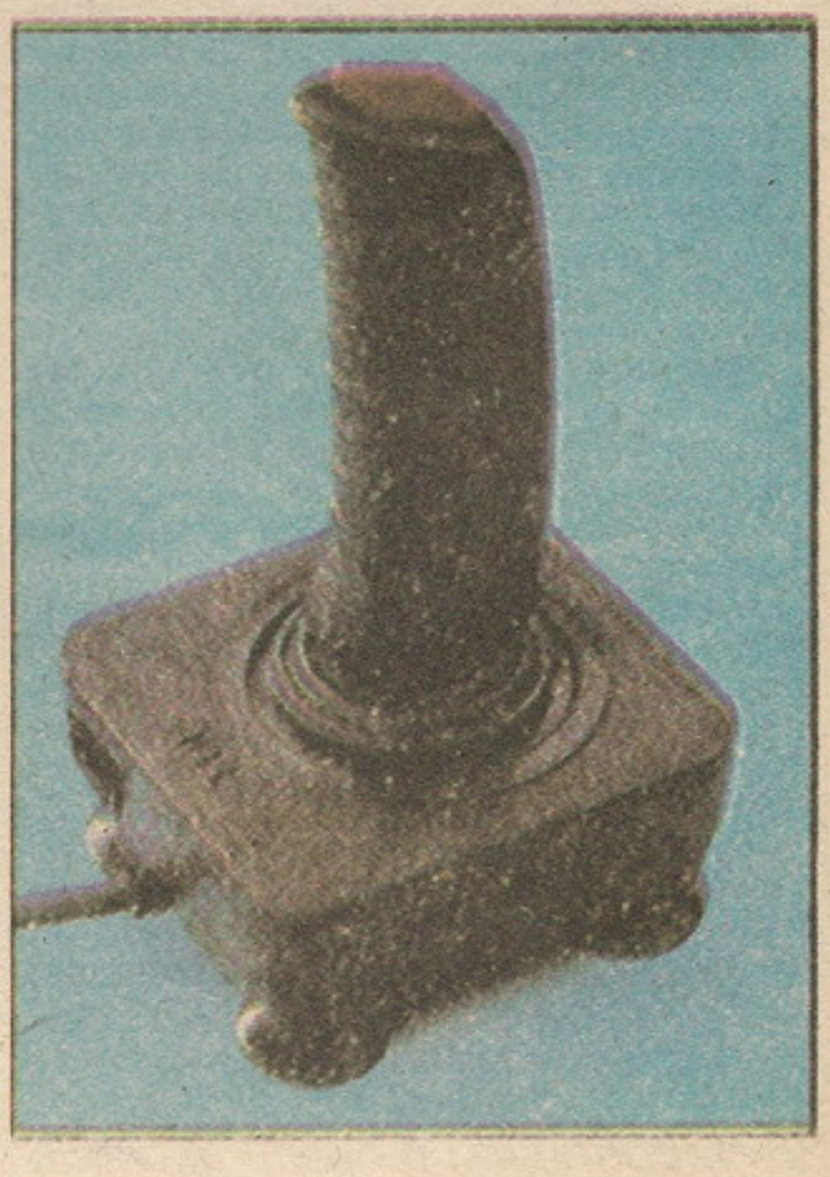
Organizacija ekrana je u mnogome preuzeta od BBC računara: postoji nekoliko grafičkih modaliteta koji nude različit broj boja u zavisnosti od željene rezolucije. U niskoj rezoluciji, svaka od 160×200 tačaka može da bude osvetljena jednom od 16 boja, dok se u visokoj rezoluciji radi sa 640×200 tačaka (izbor brojke 200 govori da Amstrad želi da plasira računare u Sjedinjenim Državama za evropske uslove „standardna" rezolucija je 640×256) i dve boje. Paleta boja za sve modove ima 28 elemenata, što znači da sam korisnik može da bira boje koje mu odgovaraju. Po uključivanju, računar se „budi" u standardnom grafičkom modalitetu koji nudi kompromis između rezolucije i broja boja: 320×200 tačaka, četiri boje i po 40 znakova u redu.

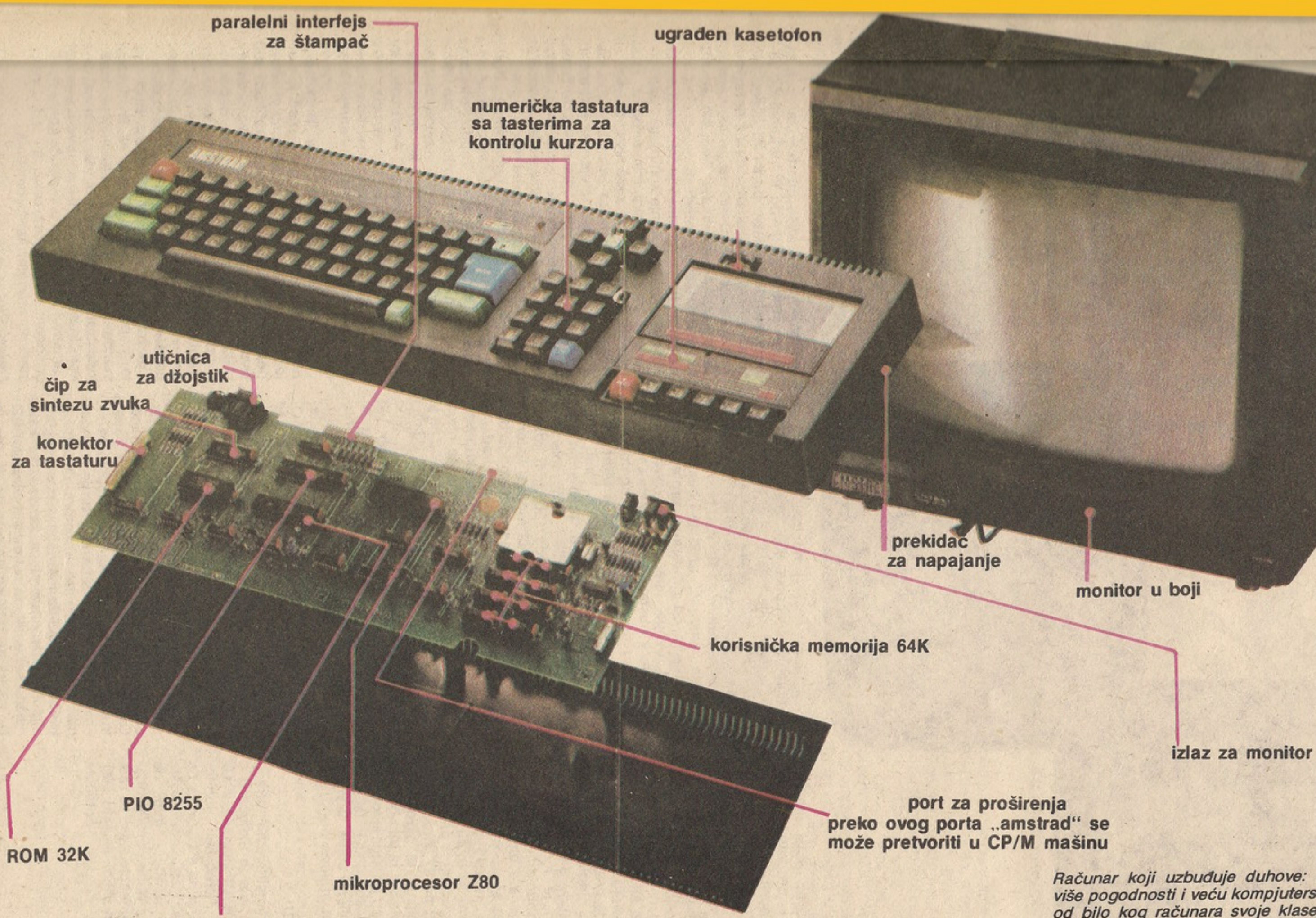
Grafička rešenja se kod CPC 464 razlikuju od onih kod Ejkornovih računara u jednoj, samo na izgled nevažnoj, pojedinosti. Dok se kod BBC i „electrona" utrošak memorije bitno menja sa promenom grafičkog modaliteta, „amstradov" ekran uvek zauzima 16 kilobajta RAM-a. Svako od ovih rešenja ima prednosti i nedostatke u odnosu na drugo: Ejkornovo rešenje je pogodnije za iskusnijeg korisnika, ali zbunjuje početnika. CPC 464, s druge strane, omogućava izbor grafičkog modaliteta koji je najpogodniji za neku primenu, a ne onoga koji je diktiran dužinom programa. Iako CPC 464 nema modalitet koji bi rezervisao samo jedan Kb za ekran, slobodan memorijski prostor je daleko pristojniji nego kod BBC računara.

Amstrad CPC 464

Tajanstvena raspodela memorije

Čuvši da CPC 464 ima 64 Kb RAM-a i 32 Kb ROM-a, mnogi će se zapitati „koliko, najzad, memorije može da ima računar sa mikroprocesorom Z80?". Ovaj procesor može da adresira najviše 64 Kb (65536 bajta), ali su konstruktori „amstrada", „orika", „epla" (isti adresni prostor može da adresira i 6502) i nekih drugih računara pribegli jednom triku: u računaru se nalaze dva paralelna bloka od po 16 Kb koji zauzimaju isti adresni prostor. Potpomognut logičkim kolima, mikroprocesor proziva onaj blok koji mu je u određenom trenutku potreban. Na ovaj način se dobija proširena memorija uz određene gubitke u brzini, koji mogu, zavisno od primene i „veštine" operativnog sistema, da budu vrlo mali ali i





Računar koji uzbuđuje duhove: lako ima više pogodnosti i veću kompjutersku snagu od bilo kog računara svoje klase, u „amstradovu" standardnu opremu uključeni su čak i kasetofon i monitor

AMSTRAD CPC 464

Mikroprocesor
Z80
Memorija
ROM 32 K
RAM 64 K pri čemu je za programe u bejziku dostupno 42 K

Ekran
Tri modaliteta sa slobodnim mešanjem teksta i grafike
640×200 (2 od 27 boja)
320×200 (4 od 27 boja)
160×200 (16 od 27 boja)

Interfejsi
palice za igru (2), „centroniks" interfejs za štampač, bas za proširenja (disk jedinice), stereo zvuk, monitor

Tastatura
profesionalna sa 74 tastera

Jezici
bejzik, paskal (na kaseti)

Dokumentacija
Uputstvo za upotrebu prilagođeno korisniku; mogu se, takođe, dobiti i bejzik i tehnički priručnik

Dimenzije
565×170×70 mm (računar sa kasetofonom)
380×350×350 (monitor u boji)

Cena
239 funti (crno-beli monitor)
349 funti (monitor u boji)

Prve igre: Kao i svaka nova mašina i „amstrad" oskudeva u programima, ali se već krajem godine očekuje prava lavina igračkog i ostalog softvera

katastrofalni. Ejkorn je želeo da mu računari budu što brži, pa ovakve trikove nije primenio, ali je ipak odškrinuo vrata za njih: čitav koncept „solidiska" koji proširuje memoriju BBC B na 256 kilobajta počiva na isključivanju i uključivanju blokova memorije. Kod „amstrada" je problem rešen već u osnovnoj verziji, pa njegovi vlasnici nemaju potrebu da kupuju skupe dodatke.

U 32 Kb smešten je moderan i brz Majkrosoftov bejzik dopunjen nekim naredbama koje su pozajmljene iz paskala. Očito je uložen veliki trud da se početniku omogući da u potpunosti iskoristi potencijale fine grafike i generatora tona, ali se u tom otišlo predaleko: omogućeno je prekrivanje ekrana sa previše (čak po 8) „prozora" za tekst i grafiku, u kojima je skrolovanje praktično nezavisno. Možda bi bilo bolje da je na ovaj način utrošen prostor iskorišćen za pojačanje matematičkog dela ROM-a, jer je tačnost funkcija prilično slaba, a vreme njihovog izračunavanja prilično dugo.

Dileme sa videom

Jedna od najjačih „amstradovih" strana je periferijska oprema koja je uključena u cenu i, uopšte, mogućnosti ekspanzije. Uz računar se isporučuje i crno-beli monitor čija je rezolucija izvanredna s obzirom na činjenicu da, zajedno sa računarom, košta 200 funti. Od takvog monitora, ipak, ne treba očekivati da bude visoko profesionalan, što se i te kako primećuje kada se radi sa 80 slova u redu. Ukoliko, umesto 200, odvojite 300 funti za računar, dobićete kolor monitor. Ovaj monitor, prema prvim utiscima iz časopisa, daje izvanredne boje, ali mu rezolucija i dalje nije jaka strana.

Prodavci kompjutera se, iako CPC 464 odlično „ide", trude da učine ovaj kompjuter interesantnim za što veći broj korisnika. Tako se „amstrad" prodaje i bez monitora ili sa nekim od proverenih profesionalnih uređaja ovoga tipa. Na taj način cena kompjutera postaje za pedesetak funti veća, ali su dobici ogromni, pogotovu ako želite da radite sa 80 slova u redu. Mlađi „hakeri" koji bi se, uglavnom, igrali mogu da kupe CPC 464 sa PAL adapterom i bez monitora. Na taj način se odriču iole finije rezolucije, ali zato štede 130 funti koje bi morali da potrošite da bi radili u koloru!

Ako se ugrađenom monitoru mogu naći zamerke, kasetofon je savršen. Uređaj je potpuno prilagođen kompjuteru (ako hoćete dobro da se zabavite, snimite na njega neku muziku, po mogućstvu opersku), pa je slušajte!), opremljen logičnim komandama i preciznim brojačem i, na posletku, lepo dizajniran. Autor ovoga teksta je uvek bio veliki pristalica specijalnih kasetofona za kompjutere i to najviše zbog kompatibilnosti softvera: ukoliko kupite kasetu sa programom za CPC 464, ili dobijete od prijatelja njegovo novo programersko remek-delo, nećete morati da se pitate „da li će se učitati?".

Sve ima svoju cenu

Osim priključka za RGB i crno-beli monitor, „amstrad" ima portove za štampač i palice za igre, kao i port za ekspanziju opšte namene. Nedostaje, očigledno, serijski interfejs, kao i port za disk jedinice. Ovakva koncepcija je pametno izabrana: kupac računara ne mora da doplaćuje za egzistencijalne potrebe kao što je štampanje (vlasnik „electrona" mora da odvoji tridesetak funti za interfejs za štampač), ali nije ni nateran da plaća za dodatke koje možda neće nikada ni koristiti (vlasnik BBC B, na kraju krajeva, ipak treba da odvoji stotinak funti za disk interfejs, iako se sa donje strane njegovog kompjutera od samog početka nalazio port nad kojim piše ‚disc drive'). Sigurni smo da će nezavisne firme naći veliki interes za prodaju manjih dodataka za „amstrada" kao što su AD konvertori i razni interfejsi.

Sama firma Amstrad je već predstavila interfejs za povezivanje kompjutera sa Hitačijevim mini disk jedinicama od 3 inča (cena i rokovi isporuke još nisu objavljeni). Nije, međutim, mnogo verovatno da će Amstrad moći da isporučuje disk jedinice u prekratkom roku, jer su glavni napori firme usmereni na proizvodnju samih kompjutera — u toku sledeće godine treba „izbaciti" preko pola miliona primeraka!

Za sada još nema mnogo komercijalnih programa za CPC 464, ali takvo stanje, po svemu sudeći, neće dugo trajati. Potrebno je određeno vreme da se prilagodi mnoštvo programa pisanih za tako popularan mikroprocesor kao što je Z80, ali posle toga nailazi prva lavina softvera. Amstradu, za sada najviše nedostaju procesori za obradu teksta i slični uslužni programi, dok prvoklasnih igara (bar se tako reklamiraju) ima dosta.

Autor ovoga teksta je imao prilike da radi sa „amstradom" koga je već kupio jedan njegov kolega. Prvi utisci su prilično nejasni: naredbe za rad sa grafikom su izvanredno realizovane, ali je matematika slaba a slika na monitoru loša. Bejzik je „ukrašen" nekim basovima koji se čudno ispoljavaju (s vremena na vreme se, na primer, na ekranu pojavi „prozor" koji uopšte nije programiran), ali je inače dobar i podržava programske strukture. Računar je, najzad, veoma brz kada se izvršavaju standardni „benčmark" testovi, ali je skoro katastrofalno spor kada se isproba na nekom inžinjerskom problemu. Sve u svemu, „amstradu" je skoro nemoguće konkurisati odnosom mogućnosti/cena, ali je to, ipak, plaćeno mnogim kompromisima.

Dejan Ristanović

prve laste donose proleće

Računari u izlogu

Electron

Računar „Electron" je, kao što je poznato, koncipiran kao oslabljena ali i dalje kompatibilna verzija računara BBC B. Kao takav, ovaj računar je prevashodno namenjen onima koji žele da učine prve korake u programiranju, upoznaju jedan kompjuter sa izvanrednim operativnim sistemom i još boljim bejzikom, savladaju osnove mašinskog programiranja i, naravno, uživaju u komercijalnim igrama. Ipak, „electronova" profesionalna tastatura i značajna brzina rada navele su mnoge ambicioznije korisnike na pomisao da bi njihov kompjuter mogao da uradi i „malo više".

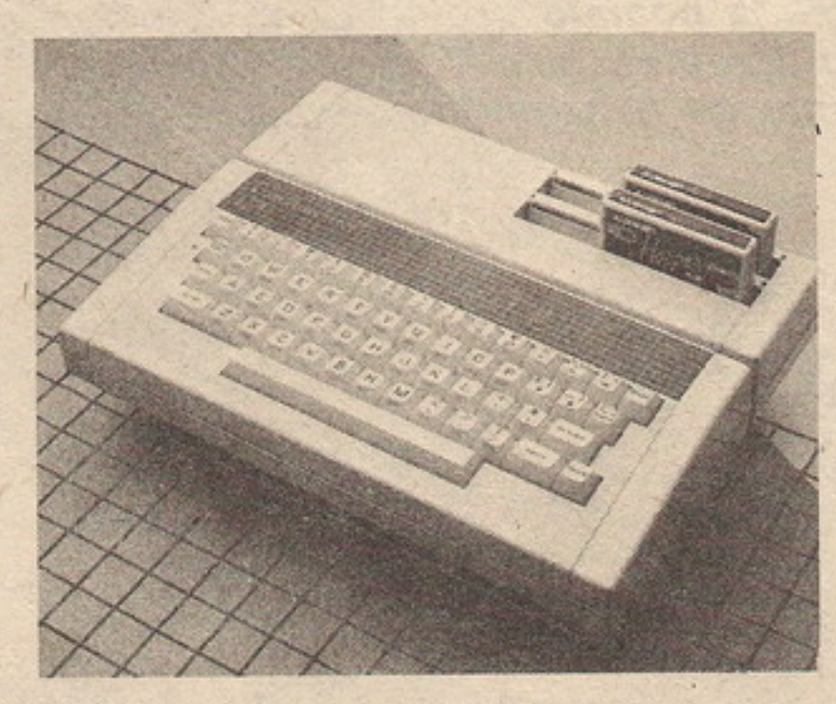
Za bilo kakvu poslovnu primenu računara neophodno mu je omogućiti da opšti sa štampačem i, u većini slučajeva, disk jedinicom. „Electron", u osnovnoj verziji, ne može da kontroliše ni jedan od ovih dodataka ne mogu da mu se priključe čak ni palice za igre! Ejkorn je brzo shvatio da „Electron" treba proširivati pa je, početkom septembra, izbacio prvi dodatak koji se, nimalo čudno, zove „Plus 1".

Interfejs za štampač

„Plus 1" je fizički prilično glomazan uređaj, čije su dimenzije jednake polovini dimenzija samoga računara. Priključuje se, prirodno, na port, ali se zbog boljeg kontakta pričvršćuje za računar posebnim šrafovima (sličnu koncepciju sledi i „interfejs 1" za „spektrum"). Priključenjem ovog dodatka „electron" dobija interfejs za standardne Centronics štampače, AD konvertor veoma pogodan za priključenje palica za igre (palice se ne dobijaju uz sam Plus 1), dva podnožja za ROM-ove sa igrama ili aplikacionim programima kao i dva porta za dalja proširenja.

Centronics interfejs nije baziran na 6522 VIA, čipu kome BBC B duguje ogromnu fleksibilnost. Ne koristi se ni neki specijalizovani ULA čip (koje Ejcorn tako rado smešta u svoje kompjutere) interfejs je, dakle, zasnovan na standardnim TTL kolima, što znači da firma ne može da proceni veličinu serije koju će tržište tražiti. EPROM od 8 Kb sadrži sav softver potreban za kontrolu štampača i drugih dodataka komande su potpuno kompatibilne sa BBC-jevima, što ne znači samo da CTRL B uključuje a CTRL C isključuje štampač već i da *FX 3 bira medij na koji će svi sistemski izlazi biti upućeni, a *FX 6 deklariše karaktere koje će štampač ignorisati. Obzirom da nije dodat RS 423 port, štampači sa lepezom i slični printeri mogu da se povezuju preko nekog od podnožja za ROM-ove (???) uz pomoć adaptera koji bi uskoro trebalo da se nađe u prodaji.

Kompatibilnost sa BBC B je ostvarena i kod AD konvertora: sve palice za igre namenjene BBC B mogu da se priključe i na „electron". Većina programa koje je Acornsoft proizveo napisano je tako da koristi tastaturu ili palice za igru, što znači da je



Most u svet; proširenje „plus 1" obezbeđuje „elekronu" najosnovnije interfejse za komunikaciju sa spoljašnjim svetom

firma od samog početka planirala „plus 1". Za ozbiljnije primene AD konvertor nudi osmobitnu rezoluciju na svakom od četiri kanala. Softver i dalje generiše brojeve iz raspona 0-65280, čime je očuvana konpatibilnost sa BBC-jevom dvanaestobitnom rezolucijom.

Kartica sa ROM-om

Mogućnost priključivanja spoljnih ROM-ova predstavlja slavni „šlager" „plusa 1". Primenjeno je rešenje koje se unekoliko razlikuje od BBC-jevog. U BBC-jeva podnožja se naime, priključuju EPROM-i 2764 i 27128. Za priključivanje svakog od njih treba, dakle, otvoriti računar, dok prečesta promena EPROM-a može da ošteti neko od ugrađenih podnožja koja su istini za volju, vrlo kvalitetna. U „plus 1" se uključuju „kartice" — male kutije u kojima se nalazi po jedan EPROM 2732, 2764 ili 27128 i specijalni konektor. Na taj način kartice se mogu menjati proizvoljno često i bez rasklapanja računara, ali zato treba izdvojiti nešto novca za kutije. Za nas se, jasno, postavlja i problem nabavke iz inostranstva.

Za sada „Ejkorn" na kartice smešta jednino igre, koje se, nakon * ROM, učitavaju pritiskom na SHIFT/BREAK ili sa CHAIN. Ukoliko kupite Snappera ili neku drugu omiljenu igru u ROM-u, oslobodićete se dosadnog upisivanja sa trake, ali i platiti mnogo više novca. Zato su ROM-ovi interesantniji kao medij za smeštanje sistemskih i aplikacionih programa. Program u ROM-u ne zauzima, naime, nikakav dodatni memorijski prostor (za vreme rada bejzik interpretator je isključen, a operativni sistem omogućava njegovo ponovno aktiviranje), pa je uz pomoć nekog monitor programa moguća odlična kontrola računara. Tekst procesor u ROM-u će biti pravo malo otkrovenje, jer će, uz interfejs za štampač, učiniti „electron" najjeftinijom mašinom na tržištu koja može da se koristi za komfornu obradu teksta (80 slova u redu, svakako, predstavlja veliku prednost „electrona" nad drugim računarima slične cene). U trenutku kada ovo pišemo, (kraj oktobra) od ozbiljnijih programa na tržištu je jedino LISP interpretator, ali je samo pitanje dana kada će se VIEW (tekst procesor) i VIEW SHEET (program za rad sa tabelama podataka, verzija odgovarajućeg programa koji je proslavio Apple II) biti prilagođeni „elekctronu".

Ne samo Ejkorn

„Plus 1", uz cenu od 60 funti, nije proširenje koga bi se trebalo lako odreći. Kao i svi drugi dodaci, međutim, „plus 1" nije bez mana. Jedna od osnovnih su interapti koje dodatak generiše usporavajući tako rad računara. Zato je korisno otkucati *FX 163 128 1 kada god nam potencijali „plusa 1" nisu neophodni i tako softverski isključiti uređaj. Neke zamerke možemo da uputimo i uputstvu za upotrebu, koje je doduše, napisano zaista izvanredno, ali samo za početnike. Iskusniji korisnici bi poželeli više informacija o izlaznim portovima koji omogućavaju priključivanje drugih dodataka, koje će, možda, poželeti sami da naprave!

Ejkorn nije jedina firma koja prodaje dodatke za „electrona". Mnogo pre „plusa 1", na tržištu se pojavio interfejs za štampač sa korisničkim portom koji proizvodi i prodaje (za 40 funti) firma Broadwau Electronics. Ovo proširenje je veoma fleksibilno zamišljeno: kupujete najpre osnovnu tablu sa konektorima pomoću kojih proključujete AD konvertor, dodatne module, disk interfejs i druge periferijske uređaje koje Brodway Electronics obećava za najbližu budućnost. Vlasnici „electrona", na taj način, mogu da pojačavaju svoj računar i približavaju se BBC-ju u onoj meri u kojoj je to potrebno, Ma koliko ga, međutim, proširivali, „electron" po brzini rada nikada neće sustići svog „starijeg brata".

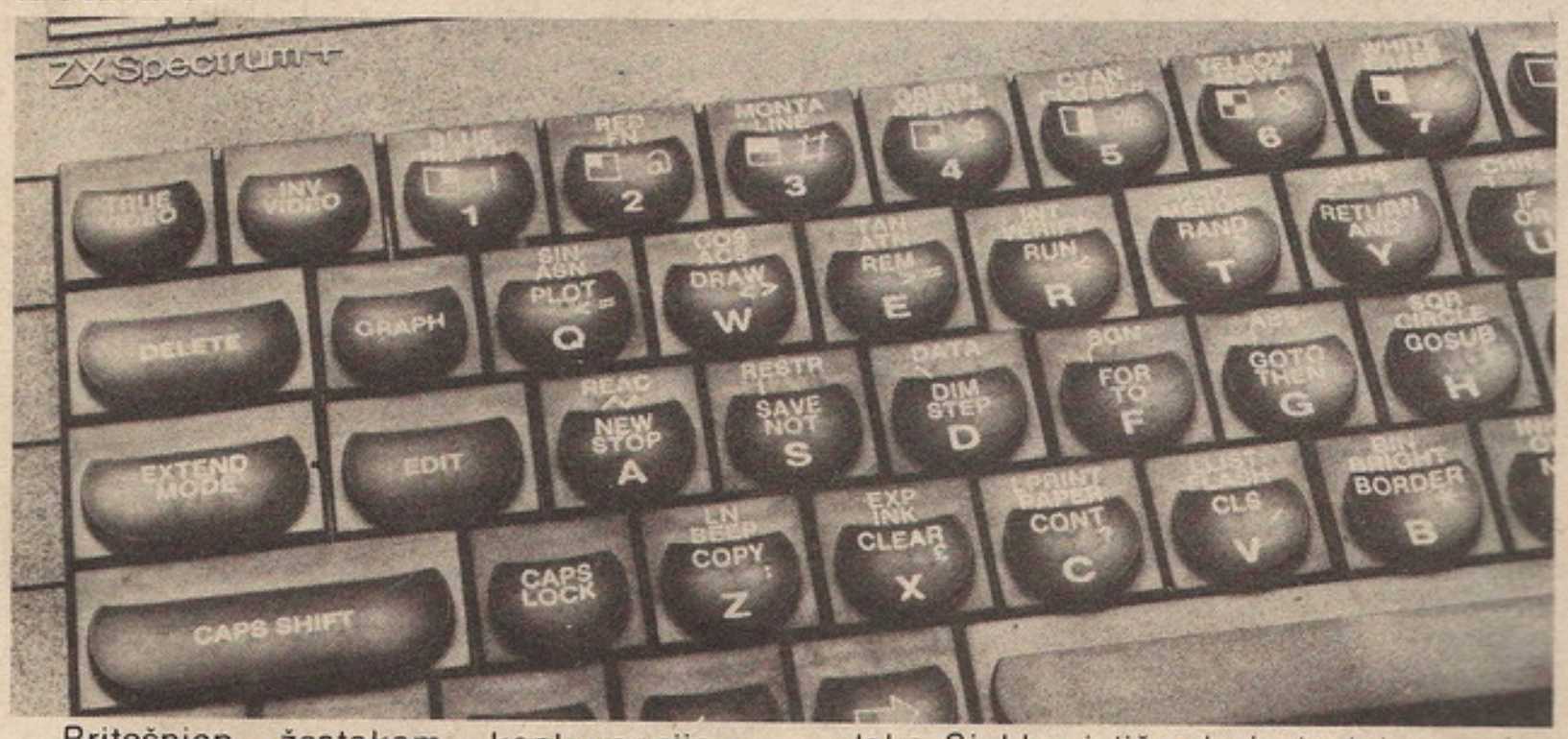
Dejan Ristanović

udarac bez odjeka

Računari u izlogu

Spektrum plus

Kada je pre više od dve godine predstavljen „spectrum“, stanje na tržištu je bilo znatno povoljnije. Bio je to prvi kolor kompjuter sa razumnom cenom, a konkurencija gotovo da i nije postojala. Stvari su postale složenije sa pojavom „komodora 64“, i, naročito, ove zime sa pojavom još dva proizvoda iz ove kuće — C16 i Plus 4, kao i „amstrada“ i MSX mašina. Kućni računari sa profi tastaturom, finom grafikom u boji i odličnim zvukom danas su jeftiniji nego ikad, i poznati „spektrum“ počinje da izgleda pomalo prevaziđeno. Sinkler je zato preduzeo hitan i, po običaju brzoplet, hirurški rez — odenuo je „spektruma“ u ruho Q1-a i opremio ga kvaziprofesionalnom tastaturom.



Pritešnjen žestokom konkurencijom, Sinkler je bio prisiljen da bira između dva sledeća rešenja: starome sprektrumu ili „spektrumu minus“ (ako se tako može nazvati) sniziti cenu ili izvršiti hirurški zahvat i dodati mu odgovarajuću tastaturu. Mnogi od nas bi više voleli da je „spektrum“ dobio pravu profi tastaturu, sa ugrađenim interfejsima 1 i 2 i, naročito, sa kompozitnim video izlazom, ali to se nije dogodilo. Bar ne za sada.

Samo nova tastatura

U stvari, Sinclair je predstavio dodatnu tastaturu za „spektrum“ i zbog toga ovaj proizvod treba ocenjivati po standardima za dodatne tastature. Na „plusu“ se nalazi više dodatnih funkcija nego na bilo kojoj drugoj tastaturi, uključujući i sve potrebne znake interpunkcije za obradu teksta.

Dodatne tipke su tačka, zarez, znaci navoda i tačka-zarez u interpunkcijskom delu i pravi i inverzni video, poništavanje (DELETE), produženi mod (EXTENDED MODE), grafički šift, šift za velika slova (CAPS SHIFT) i PAUZA. Postoje i dodatni CAPS SIFT i SYMBOS SHIFT, zajedno sa simpatičnom razmaknicom između četiri tipke za kontrolu kursora.

Iako Sinkler ističe da je tastatura „plusa“ delo ergonomskih genija, mnogi se neće složiti sa njim: tačka je na manje-više uobičajenom mestu, ali su zarez, tačka-zarez i navodnici gurnuti u donji red, upravo ispod CAPS SHIFTA. IBM je pokušao da ovo proturi na svojoj tastaturi kao industrijski standard, Sinclair je to prihvatio, no većina ostalih proizvođača mikroa smatra da je bolje staviti šift tipke u poslednji red. Sve u svemu, ljudima je već dosta egzotičnih rasporeda interpunkcijskih tipki.

Sa stanovišta programera, raspored je mnogo razumniji. Prozaični trgovci mogu da negoduju što je SYMBOL SHIFT upravo ispod CAPS SHIFTA, ali ovo umnogome olakšava generisanje znakova pod šiftom (velikih slova) i komandi. Takođe, šiftovi za grafički i prošireni mod, kao i EDIT i DELETE tipke, nalaze se na zgodnim mestima. Pauza je pomalo zagonetna, pošto pored posebne tipke funkcioniše i stari sistem (CAPS SHIFT i razmaknica).

Korisna novina je što slučajni laki pritisak na pauzu nema nikakvog efekta.

Ipak ne bez šansi

Konstrukcija tipki je slična kao na QL tastaturi. Poklopac tipke pritiska prekidač, a ovaj membranu ispod sebe. Osećaj na manjim tipkama nije loš. Međutim, veće imaju samo jedan prekidač i mogu prilično neugodno da se naginju sa jedne na drugu stranu. Kupice tipki su dizajnirane kao na QL-u, ali pošto na svakoj tipki ima mnogo više natpisa, ukupni utisak je da je tastatura prilično nepregledna.

Tastatura, kao ni na QL-u, nije zakočena, ali se to postiže pomoću dve ispupčene nožice na zadnjim uglovima, koje naginju celu tastaturu prema napred, ostavljajući dovoljno mesta ispod za interfejs 1. Priključiti interfejs 2 je sasvim drugačiji problem — ako su nožice izvučene, on visi u vazduhu, pa priključivanje karica sa ROM-om može biti opasno.

Velika novost je i reset taster, postavljen na levoj strani kućišta. Govorilo se i o reviziji štampane ploče, ali, izgleda, postoje samo manje razlike između ploče na „plusu“ (verzija već 4b!) u odnosu na postojeću verziju 3.

Uzimajući u obzir cenu, „spektrum plus“ je dobro zašao u tržište Komodora. U poređenju sa „64“ i „atmosom“, novi „spektrum“ nije bez šansi, ali je još uvek daleko od toga da nadvisi za klasu svoje protivnike.

Iako Sinkler želi da dokaže da je ova vrsta tastature podjednako dobra kao i profesionalna (sa dugim hodom), mnogi misle da nije, i krajnje je vreme da Sinclair Research uzme u tim jednog daktilografa i proizvede neproblematičnu tastaturu. To bi svim njegovim mašinama donelo istinski plus.

Da jeste kao što nije

Da bi se „spektrum plus“ dobro i prodavao, morao bi da bude jeftiniji ili bolji. Ne vidi se ekonomsko opravdanje da nova tastatura košta 50 funti; sa cenom od 130—140 funti, „plus“ bi bio siguran pobednik na tržištu u svojoj klasi. Druga mogućnost je (o čemu se već pronose glasine) da se prestane sa proizvodnjom „spectruma minus“ (sadašnje verzije), spusti cena „plusu“ na 130 funti, i izbaci nova mašina za oko 190 funti — Spectrum Squared (na kvadrat). Ona će imati ugrađen interfejs 1, centroniks interfejs i, možda, pravu tastaturu?

Kada je u Sinclair Research-u razgovarano o „plusu“, bilo je predloga da se pojavi sa ugrađenim interfejsom i utičnicom za karticu. Ovaj predlog je odbijen, ali nije definitivno napušten. Šifrovano ime za projekt „Spectruma Plus“ je bilo „Thunderbird 1“, a kao što pratioci TV serija znaju, bilo je 5 Thunderbirda.

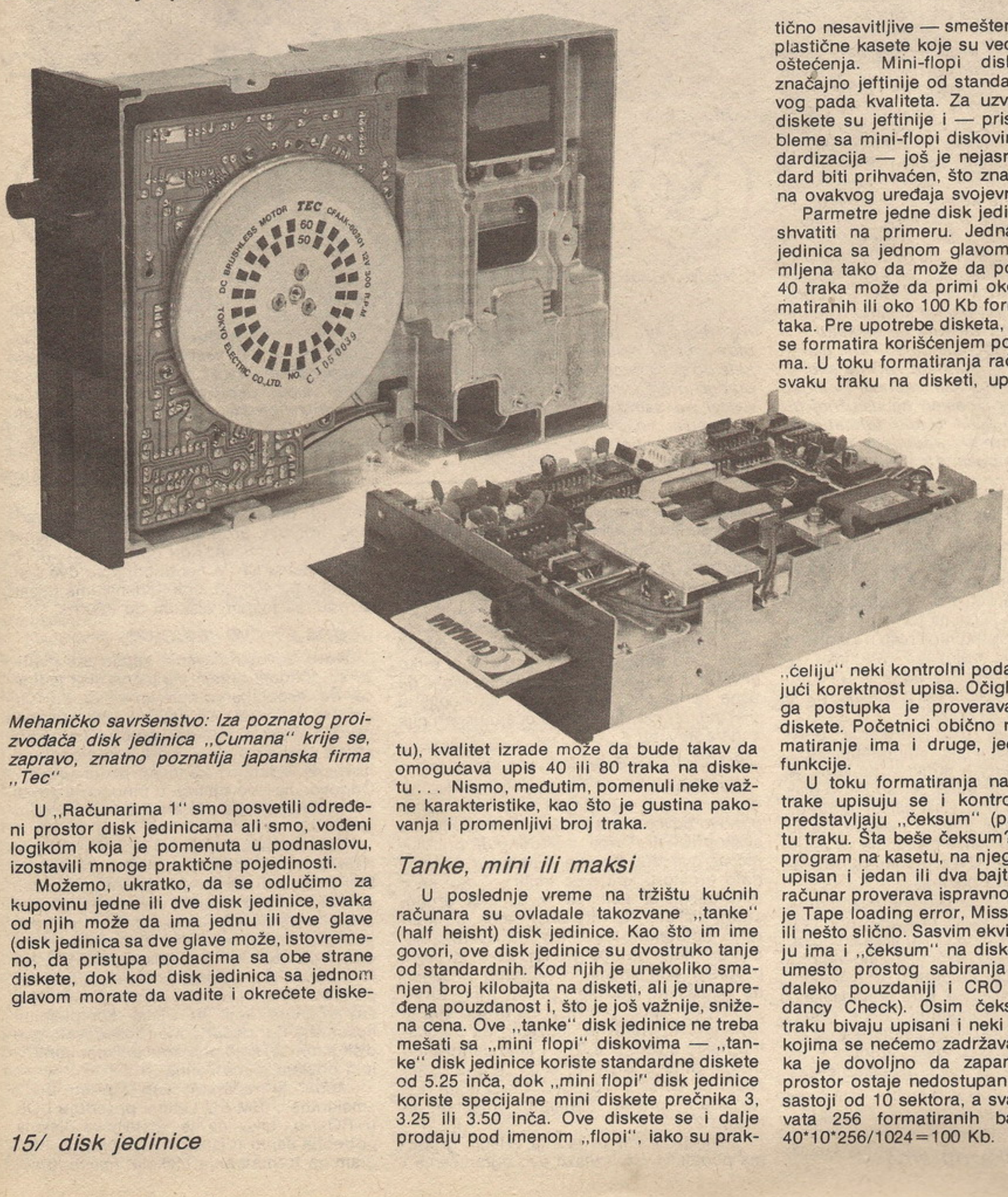
Sam Sinkler izjavljuje da on nije napustio ideju daljeg dograđivanja „spektruma“. i nagoveštava dugoročni projekat — potpuno prenosivu mašinu sa ugrađenim disk jedinicama i displejom sa ne-LCD tehnikom (što sugeriše rešenje sa ravnim TV ekranom). Ne treba, međutim, očekivati da će se ovakva mašina pojaviti u sledećih dvanaest meseci. Ostalo je nejasno da li će to biti „spektrumova“, QL-ova tehnologija ili nešto sasvim novo!

Miodrag Klajić

disk jedinice

Periferijska oprema

Disk jedinice su sve do skora predstavljale elitnu i za naše džepove relativno nedostupnu periferijsku opremu računara. Svakodnevnim napretkom tehnologije cene disk jedinica padaju tako da su u ovom trenutku postale pristupačne čak i onom programeru koji ne namerava da se bavi nekim velikim biznisom. Koji tipovi disk jedinica postoje i kako izvršiti najbolji izbor?



Mehaničko savršenstvo: Iza poznatog proizvođača disk jedinica „Cumana" krije se, zapravo, znatno poznatija japanska firma „Tec"

U „Računarima 1" smo posvetili određeni prostor disk jedinicama ali smo, vođeni logikom koja je pomenuta u podnaslovu, izostavili mnoge praktične pojedinosti.

Možemo, ukratko, da se odlučimo za kupovinu jedne ili dve disk jedinice, svaka od njih može da ima jednu ili dve glave (disk jedinica sa dve glave može, istovremeno, da pristupa podacima sa obe strane diskete, dok kod disk jedinica sa jednom glavom morate da vadite i okrećete disketu), kvalitet izrade može da bude takav da omogućava upis 40 ili 80 traka na disketu . . . Nismo, međutim, pomenuli neke važne karakteristike, kao što je gustina pakovanja i promenljivi broj traka.

Tanke, mini ili maksi

U poslednje vreme na tržištu kućnih računara su ovladale takozvane „tanke" (half heisht) disk jedinice. Kao što im ime govori, ove disk jedinice su dvostruko tanje od standardnih. Kod njih je unekoliko smanjen broj kilobajta na disketi, ali je unapređena pouzdanost i, što je još važnije, snižena cena. Ove „tanke" disk jedinice ne treba mešati sa „mini flopi" diskovima — „tanke" disk jedinice koriste standardne diskete od 5.25 inča, dok „mini flopi" disk jedinice koriste specijalne mini diskete prečnika 3, 3.25 ili 3.50 inča. Ove diskete se i dalje prodaju pod imenom „flopi", iako su praktično nesavitljive — smeštene su u robusne plastične kasete koje su veoma otporne na oštećenja. Mini-flopi disk jedinice su značajno jeftinije od standardnih bez ikakvog pada kvaliteta. Za uzvrat, standardne diskete su jeftinije i — pristupačnije. Probleme sa mini-flopi diskovima pravi i standardizacija — još je nejasno koji će standard biti prihvaćen, što znači da je kupovina ovakvog uređaja svojevrstna kocka.

Parmetre jedne disk jedinice najlakše je shvatiti na primeru. Jedna „tanka" disk jedinica sa jednom glavom koja je pripremljena tako da može da podeli disketu na 40 traka može da primi oko 125 Kb neformatiranih ili oko 100 Kb formatiranih podataka. Pre upotrebe disketa, naime, mora da se formatira korišćenjem posebnog programa. U toku formatiranja računar proverava svaku traku na disketi, upisujući u svaku „ćeliju" neki kontrolni podatak i proveravajući korektnost upisa. Očigledna svrha ovoga postupka je proveravanje ispravnosti diskete. Početnici obično ne znaju da formatiranje ima i druge, jednako značajne funkcije.

U toku formatiranja na početak svake trake upisuju se i kontrolni podaci koji predstavljaju „čeksum" (proveru zbira) za tu traku. Šta beše čeksum? Kada upisujete program na kasetu, na njegovom kraju biva upisan i jedan ili dva bajta pomoću kojih računar proverava ispravnost upisa i ispisuje Tape loading error, Missed data, WHAT? ili nešto slično. Sasvim ekvivalentnu funkciju ima i „čeksum" na disketi, s tim što se, umesto prostog sabiranja bajtova, koristi daleko pouzdaniji i CRO (Cyclic Redundancy Check). Osim čeksuma, uz svaku traku bivaju upisani i neki drugi podaci na kojima se nećemo zadržavati — za korisnika je dovoljno da zapamti da određen prostor ostaje nedostupan. Svaka traka se sastoji od 10 sektora, a svaki sektor obuhvata 256 formatiranih bajtova, što čini 40*10*256/1024=100 Kb.

KAKO IZGLEDA DISKETA?

Na slici 1 shematski je prikazana jedna disketa. Osim dimenzija u milimetrima i inčima vidimo da je površina preko koje računar upisuje i čita podatke ograničena na mali prozor, dok je ostatak diskete zaštićen čvrstim kartonskim omotačem. Ako želite da saznate zašto se diskete označavaju i kao „fleksibilni (savitljivi) diskovi", presecite ovaj omotač i uzmite u ruke „suštinu" diskete koju ste, jasno, upropastili.

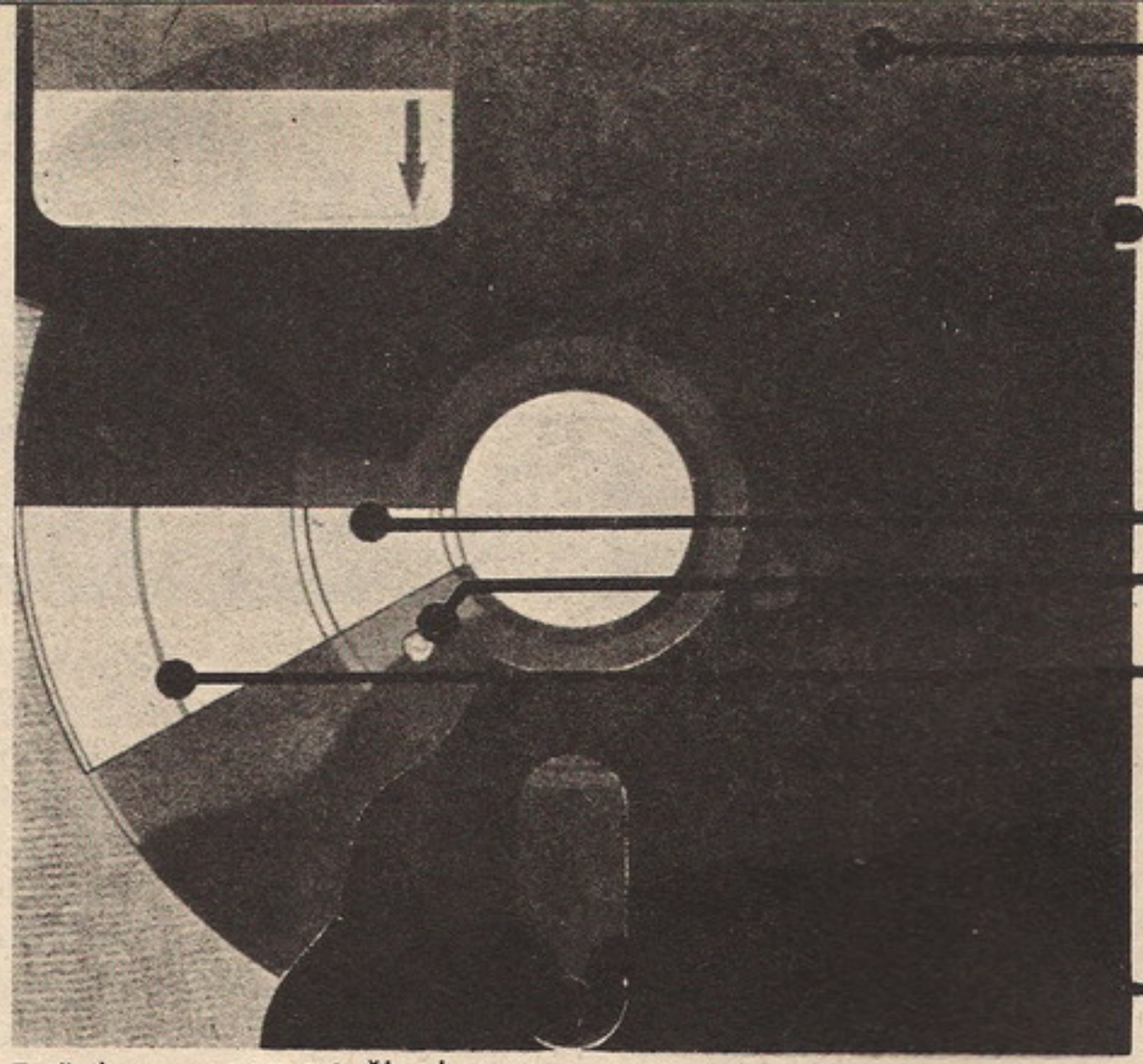
Osim otvora za pristup glave i centralnog otvora koji omogućava okretanje, disketa poseduje i tzv. indeksni otvor koji, kao što smo objasnili u „Računarima 1", omogućava računaru da locira prvi sektor svakog traga.

Sa leve strane diskete se nalazi zarez koji pokrivate posebnom nalepnicom kada poželite da zaštitite programe i podatke na njoj od slučajnog brisanja. Ova nalepnica odgovara lomljenju pipaka sa donje strane magnetofonske kasete, ali obratite pažnju na to da je situacija upravo obrnuta: ne može se upisivati na kasetu koja nema pipke i na disketu koja ima nalepnicu.

Disketa se čuva u papirnoj omotnici u koju se uvek smešta tako da je prozor za glavu zaštićen papirom. Sa gornje strane se nalazi mesto za nalepnicu na kojoj možete da napišete ime diskete i podatke o programima na njoj. Uvek najpre ispišite nalepnicu pa je tek onda zalepite na disketu!

Pre nego što počnete da koristite disketu morate da je formatirate kao na slici 2. Formatiranje obavlja sam računar pomoću programa koji je smešten na uslužnoj disketi ili upisan u ROM. U toku formatiranja disketa biva softverski podeljena na 40 ili 80 koncentričnih traka, a svaka od tih traka na 10 sektora. Nulta traka se uvek nalazi bliže obodu diskete i sadrži ime diskete, katalog programa koji su upisani na nju i mnogo drugih informacija. Između sektora nalazi se sadržaj koji 99% korisnika ne primećuje: bajtovi za sinhronizaciju i čeksumi koji omogućavaju da se utvrdi eventualna greška ili oštećenje.

Uspešno formatiranje garantuje ispravnost diskete i pokazuje da će ona, ukoliko ne bude mehanički oštećena, služiti praktično večito. Ukoliko pri formatiranju računar prijavi grešku, najbolje je baciti disketu — rizik od gubitka vrednih programa i podataka je preveliki. Ovakve greške se, na sreću, vrlo retko javljaju.



Pažnja, opasno po život:

Četrdeset ili osamdeset traka

Pođimo korak dalje i omogućimo našoj disk jedinici da na svaku disketu upisuje 80 traka sa podacima. Time smo, jasno, povećali kapacitet na 200 Kb. Mnogi početnici misle da 40 traka zauzima samo pola diskete i da jeftinije disk jedinice nisu u stanju da „dosegnu" drugu polovinu. Nije tako — kod skupljih disk jedinica glava je u stanju da se dvostruko preciznije postavlja, pa je isti prostor bolje iskorišćen.

Postojanje ova dva standarda pravi određene probleme sa kompatibilnošću. Većina komercijalnog softvera se prodaje na disketama sa 40 traka, jer su takve disk jedinice do skora bile u isključivoj upotrebi. Sada, jasno, mnogi kupuju diskove većeg kapaciteta i imaju velike probleme sa zaštićenim programima. Ovi problemi se rešavaju na jedan od dva načina.

Većina proizvođača prodaje disk jedinice na kojima se nalazi prekidač pomoću koga korisnik bira mod sa 40 ili sa 80 traka. Ove disk jedinice su obično znatno skuplje od disk jedinica bez prekidača pri jednakim drugim karakteristikama i ne predstavljaju dobru investiciju. Moguće je, naime, sastaviti program pomoću koga se disketa od 40 traka učitava pomoću disk jedinice koja normalno radi sa 80 traka, pri čemu svaka druga traka biva preskočena. Ovakav program se, kod boljih disk interfejsa, ugrađuje u standardni DOS (npr. kod Watford disk interfejsa za BBC B).

Nijedno od ovih rešenja nije potpuno: omogućava učitavanje softvera koji je snimljen na četrdesetotračnoj disk jedinici, ali ne i snimanje programa u formatu od 40 traka. Ovo snimanje, što proizvođači retko ili nikada ne govore u reklamama, može da uspe samo ako snimate na tek kupljenu disketu, ali čak ni u tom slučaju uspeh nije zagarantovan!

Jedna ili dve glave

Proširujući našu zamišljenu disk jedinicu, dodaćemo joj drugu glavu. Tako ćemo ponovo udvostručiti kapacitet — 400 Kb na svakoj disketi. Treba, međutim, da primetimo da ovih 200 Kb nije jednostavno pridodato prostoru koji smo imali — radi se o informacijama koje se nalaze na drugoj strani diskete. Ne možemo, dakle, da kreiramo datoteku od 400 Kb već, eventualno, dve datoteke po 200 Kb. Isto tako, za svaki program koji snimamo moramo posebno da naglasimo gde se upućen (npr. SAVE „2.PROG" upućuje PROG na drugu stranu diskete). Alternativno, možemo da otkucamo nešto poput DRIVE 2 i tako definišemo drugu stranu diskete kao „radnu", što znači da će se svi programi kod kojih nije naglašen „pravac kretanja" smeštati na stranu 2.

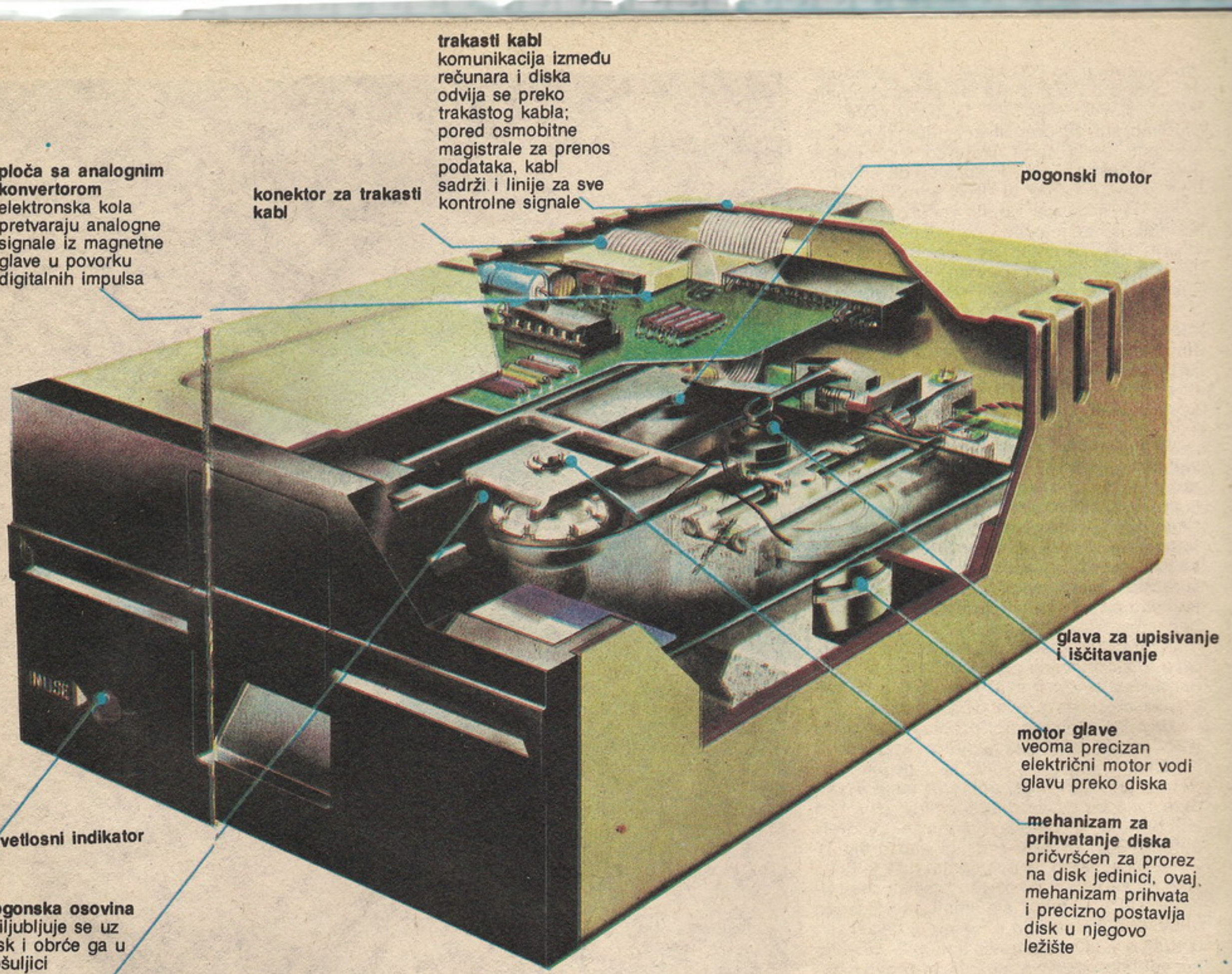
Bolji programi za rad sa velikim grupama podataka prevazilaze ovo ograničenje i stvaraju utisak da posedujete 400 Kb radnog prostora, program sam „misli" o tome šta se gde smešta. Kod čuvenog operativnog sistema CP/M neophodne su dve disk jedinice po 400 Kb, dok korisnik ima utisak da radi sa jednim diskom od 800 K!

Jedna ili dve jedinice

Novo udvostručavanje kapaciteta postiže se dokupljivanjem još jedne disk jedinice. Na raspolaganju nam je, dakle, 2×100, 2×200 ili 2×400 Kb koji su „rascepkani" na dve diskete i (eventualno) na po dve strane svake od njih. Ima li svrhe duplirati novčane izdatke i kupovati drugi uređaj? Odgovor na ovo pitanje u mnogome zavisi od računara i disk interfejsa koji posedujemo.

Neki računari (npr. TR8 80) drže u ROM-u samo mali deo DOS-a (programa koji podržava rad sa disk jedinicama), dok ostatak nerekidno učitavaju sa takozvane „sistemske diskete" koja je stalno umetnuta u prvi drajv. Ukoliko imamo samo jedan drajv, pa još ako on ima samo 100 Kb formatiranog prostora, za naše programe i podatke ostaje premalo mesta. Osim toga, za računare koji su CP/M kompatibilni apsolutno je neophodno posedovati dve disk jedinice, koje pod softverskom kontrolom postaju jedna celina.

Moderniji računari, kao što su BBC, „mekintoš", IBM PC i slični poseduju DOS u ROM-u, tako da je sistemska disketa potrebna samo u izuzetnim prilikama. Program za formatiranje disketa, naime, obič-



no nije smešten u ROM, jer je njegov slučajan poziv često fatalan — formatirajući disketu možete nehotice da obrišete sve podatke na njoj! Kod ovih računara obično radimo sa jednom disk jedinicom, pa bi druga bila luksuz da ne postoji potreba za kopiranjem.

Svaki bolji disk interfejs poseduje naredbu BACKUP pomoću koje kopiramo svaki bajt neke diskete na drugu, praveći tako identične kopije. Obzirom da smo uvek govorili o pouzdanosti disk jedinica, moramo da odgovorimo na pitanje koje se samo nameće svakom čitaocu ovih redova: da li je ovo „kloniranje" disketa uopšte potrebno?

Disk jedinice su zaista izuzetno pouzdane, ali su diskete relativno lako kvarljiva roba. Dovoljno je da nam disketa ispadne na patos (ponekad čak i na sto), pa da se pod njen omotač uvuče nešto prašine, koja može da učini neki bajt nečitljivim. Nevolja sa disk jedinicom je u tome što jedan nečitljiv bajt može da učini čitavu disketu neupotrebljivom (većina disk interfejsa će odbijati da opšti sa disketom na kojoj su „čeksumi" pogrešni). Po Marfijevom zakonu (verovatnoća da nešto pođe naopako proporcionalna je sa njegovim značejem), problem sa nečitljivošću diskete će uvek nastupiti u najneprijatnijem mogućem trenutku — kolika je šteta ako vam nekoliko meseci rada propadne baš pred sam njegov završetak? Zato se, u većim firmama, diskete kloniraju svako veče, i to u nekoliko primeraka, čime se praktično eliminiše verovatnoća greške. Ovaj tekst, na primer, radimo na programu za obradu teksta, pri čemu se rezultat upisuje na tri diskete: jedna verzija na prvu, sledeća na drugu, pa na treću, i onda opet na prvu. Tako, u najgorem slučaju, može da se desi da propadne poslednja verzija teksta, dok su dve prethodne sačuvane.

Kada se dođe do kopiranja diskete, rad sa jednim drajvom se pretvara u pravo mučenje. Kućni računari, naime, retko imaju 400 Kb RAM-a. Ovolika memorija bi omogućila da sa kompletan sadržaj diskete upiše u RAM, da korisnik zameni disketu i da, onda, računar na nju upiše odgovarajući sadržaj. Ukoliko, međutim, vaš računar ima 32 Kb memorije, moraćete naizmenično da menjate dve diskete i to dvadesetak puta da bi prepisali sadržaj jedne na drugu, što je posao od petnaestak minuta! Pokušajte da ovo radite svakodnevno, pa ćete za deset dana (?!) kupiti drugi disk koji će ovaj posao automatizovati!

Kupujući drugu disk jedinicu, obično se opredeljujemo za istovetan tip, mada nema nikakve prepreke da, na primer, kupimo

Disk jedinice u 200 reči

Disk jedinica se sa kućnim računarom povezuje preko posebnog interfejsa koji najčešće treba dokupiti.

Možete da se odlučite za jednu, dve ili (retko) više disk jedinica koje će biti povezane sa vašim računarom. U stranoj literaturi se uređaj koji sadrži dve disk jedinice ('dva drajva') naziva 'dual unit'.

Svaki drajv može da ima jednu ili dve glave omogućavajući upis podataka sa jedne ili dve strane diskete. Upis sa obe strane (u stranoj literaturi 'double side') udvostručuje raspoloživu količinu informacija uz određeno povećanje cene.

Na disketu može da bude upisano 40 ili 80 traka sa podacima. Svaka traka se, pri standardnoj gustini upisa, sastoji od 10 sektora po 256 bajta.

Neki interfejsi omogućavaju dvostruku gustinu upisa ('double density'), povećavajući kapacitet diskete za oko 60%.

Jednostruka jednostrana disk jedinica sa 40 traka omogućava „pamćenje" 100 Kb informacija i košta ispod 100 funti. Maksimalni kapacitet dvostrukog dvostranog drajva sa dvostrukom gustinom upisa iznosi 1.5 Mb, a cena retko prelazi 500 funti.

U poslednje vreme se eksperimentiše sa masivnim ('hard') diskovima kapaciteta 5 — 50 Mb. Jedan od najpopularnijih tipova je Vinčester koji, zavisno od karakteristika, košta 1500 do 5000 funti i obično je kombinovan sa nekim drugim proširenjima računara.

disk jedinicu od 200 Kb iako već posedujemo jednu od 400 K. Na taj način uštedimo nešto novca (oko 30 funti) i ništa ne gubimo što se prepisivanja tiče. Čini nam se, ipak, da 30 funti nije vredno problema koji će nastupiti ako pokušamo da učinimo naš računar CP/M ili UNIX kompatibilnim. Uvek je moguće kupiti računar slabijih karakteristika, a zatim ga postepeno dograđivati. Disk jedinica se, međutim, kipuje jednom za svagda i retko je koja firma raspoložena da zameni disk jedinicu za kvalitetniju uz doplatu: i najbolje disk jedinice su danas toliko jeftine da je svaka firma srećna kada iz magacina izbaci sve jeftinije!

Sa napajanjem ili bez njega

Pri kupovini druge disk jedinice postavlja se i problem napajanja. Većina novijih računara je projektovana tako da njihov ispravljač zadovoljava potrebe samog računara, jedne disk jedinice i nekih manjih proširenja. Zato je logično kupiti prvi disk bez ispravljača i stabilizatora napona, jer se tako štedi dvadesetak funti. Za drugu disk jedinicu potrebno je, obično, kupiti i poseban izvor za napajanje, a tada je logično rasteretiti računarev ispravljač napajanja prve disk jedinice. Sve to zahteva malu hardversku intervenciju koja nekim početnicima može da predstavlja problem.

Napajanje nije jedini problem koji treba rešiti da bi dve disk jedinice kupljene u različito vreme proradile kao celina. Potrebno je, osim toga, pomeriti neke džampere tako da svaka disk jedinica „zna" na koji broj da se odaziva (obično se prvi drajv definiše kao drajv 0 a drugi kao drajv 1). Osim toga, samo na poslednjoj disk jedinici sme da se nalazi takozvani „resistor terminator set". Ukoliko ste imali samo jedan drajv, RTS se nalazio u njemu, a nalazi se i u drajvu koji ste upravo dokupili. Ukoliko posedujete uputstva, nećete se previše namučiti. Zbog problema sa carinom, međutim, uputstva često ostanu u nekom nemačkom košu, a tada nastupaju pravi problemi.

Postoji još jedan način da praktično duplirate kapacitet vaše diskete, ali on ne zavisi od disk jedinice, već od računara i disk interfejsa.

Jednostruka ili dvostruka gustina

Mnogi smatraju da „double density" (dvostruko guste u bukvalnom prevodu) diskete imaju 160 umesto 80 traka. Ništa pogrešnije od toga — ovakvo pakovanje može da se primeni kako na disketama od 80 traka tako na disketama od 40 traka, pa čak i na sada sasvim zastarelim disketama od 35 traka. Pri tome se količina informacija koja se smešta na disketu praktično udvostručuje, pri čemu vreme pristupa, čak, opada! Ukoliko, na primer, imamo disk jedinicu od 400 Kb i kupimo „double density" disk interfejs, radićemo sa oko 720 Kb formatiranog prostora!

Odakle ovaj dobitak? Ništa jednostavnije: način pakovanja podataka je rešen inteligentnije i — to je sve. Disk interfejs ovoga tipa je zasnovan na drugim, savremenijim čipovima, i apsolutno je nekompatibilan sa disketama snimljenim standardnom gustinom. Neki bolji interfejsi rešavaju problem

Pouzdana ali spora: Sa disk jedinicom vlasnici računara „komodor 64" nemaju naročitih problema — osim što je toliko spora da se u brzini učitavanja sa njom može meriti čak i najobičniji kasetofon

kompatibilnosti ugrađivanjem posebnog programa koji simulira „zastarele" čipove, ali je takvo rešenje polovično — neprimenljivo za zaštićene komercijalne programe.

I pored očitih prednosti, do sada se još nije prešlo na „double density" interfejse. Razlozi, verovatno, leže u činjenici da je 400, odnosno 800 Kb sasvim dovoljno za sve primene i da je sledeća stepenica masivni (hard) disk od dvadesetak mesabajta. Drugi razlog je u tome što bi ova gustina pakovanja mogla da zastari daleko pre nego što bude prihvaćena — već se postižu lepi rezultati sa takozvanom „četvorostrukom gustinom" pakovanja podataka, koja čini da na disketu koja je ranije primala 400 Kb stane megabajt i po!

Disk jedinice . . .

Kupujući disk jedinicu, obraćate se posredniku koji, zapravo, jedino pakuje ove precizne uređaje u kutije, piše na njih svoju firmu i prodaje sistemsku disketu i sve kablove potrebne za priključivanje na vaš kompjuter. Same disk jedinice su obično TEC-ove, Mitsubishi-jeve i Canonove. Japanci drže pun monopol na proizvodnju „tankih" disk jedinica za Apple, BBC i druge kućne računare.

Cumana je ime jedne od najjačih firmi koja prodaje kako TEC-ove tako i Mitsubishi-jeve disk jedinice pa ćemo razmotriti njihovu liniju. Cumana prodaje uređaje iz serija CS, CSX, CSE, CD i CD/S.

• CS (Cumana Singl) serija obuhvata jednostruke jedinice sa izvorom za napajanje i potpunom dokumentacijom u koju su uračunate i sistemske diskete. CS100 košta 169, CS200 — 209, a CS 400 celih 233 funte. Ovo su izvozne cene na koje, ako kupujete u Engleskoj, morate da platite 15% VAT („Vallue add tax"); ovaj iznos bi trebao da vam bude vraćen uz potvrdu da ste uređaj izneli iz Engleske u određenom roku.

• CSX serija obuhvata jednostruke disk jedinice sa kompletnom dokumentacijom i sistemskim disketama, ali bez izvora za napajanje. Moraćete, dakle, da iskoristite ispravljač vašeg računara, ili da sami napravite potreban izvor od 5 odnosno 12 V. Cene za modele CSX 100, CSX 200 i CSX 400 su redom 149, 193 i 215 funti. Pre nego

Naziv	Firma	Tip	Traka	Strana	Kapacitet	Cena
Slimline	Twillstar	Jednostruki	40	1	100	147
Slimline	Twillstar	Dvostruki	40/80	1	400	287
Slimline	Twillstar	Dvostruki	40/80	2	800	486
Slimline	Twillstar	Jednostruki	80	2	400	269
Slimline	Twillstar	Dvostruki	80	2	800	503
	AMS	Jednostruki	40	2	200	195
	AMS	Dvostruki	40	2	400	347
FD55A	Viglen	Jednostruki	40	1	100	135
FD55E	Viglen	Jednostruki	40/80	1	200	169
FD55B	Viglen	Dvostruki	40	2	400	400
FD55F	Viglen	Dvostruki	40/80	2	800	416
	Carson Dev.	Jednostruki	40	1	100	140
	Carson Dev.	Dvostruki	40	2	200	170
	Carson Dev.	Dvostruki	80	2	800	400
PSD1	Pace	Jednostruki	40	1	100	149
PSD3	Pace	Jednostruki	40/80	2	400	256
PDD1	Pace	Dvostruki	40	1	200	294
PDD3	Pace	Dvostruki	40/80	2	800	417
	Micro Res.	Jednostruki	45	1	100	130
	Micro Res.	Dvostruki	45	1	200	216
CS100	Cumana	Jednostruki	40	1	100	169
CD200	Cumana	Dvostruki	40	1	200	305
CD400/S	Cumana	Dvostruki	40/80	1	400	469
CD800/S	Cumana	Dvostruki	40/80	2	800	499
	Watford	Jednostruki	40	1	100	169
	Watford	Jednostruki	40/80	2	400	215
	Watford	Dvostruki	80	2	800	499
	Technomatic	Jednostruki	40	1	100	150
	Technomatic	Dvostruki	40/80	1	400	400
	Technomatic	Dvostruki	80	2	800	420
	Midwich	Jednostruki	40	1	100	153
	Midwich	Dvostruki	40/80	1	400	479
	Opus	Jednostruki	40	1	100	149
	Opus	Dvostruki	40/80	1	400	514
	Opus	Dvostruki	80	2	800	430

(1) Sve cene su u funtama sa uključenim VAT.
(2) Svi kapaciteti su i kilobajtima.

što se odlučite da opteretite vaš računar novim uređajem, raspitajte se da li on može da dâ potrebnih 9 W.

• CSE serija obuhvata disk jedinice sa izvorom za napajanje ali bez ostale dokumentacije i sistemskih disketa. Ove uređaje kupujete ako već posedujete uređaj iz serije CS ili SCX i želite da duplirate svoje disk-potencijale. Pridodati su potrebni kablovi i uputstvo za povezivanje. Cene za modele CS100E, SC200E i CS400E su redom 153, 195 i 219 funti.

• CD (Cumana Double) serija obuhvata, jasno, dvostruke disk jedinice (dva drajva u jednoj kutiji) uz svu dokumentaciju i zajedničko napajanje. CD 200 košta 305, CD400 — 386 a CD 300 celu 431 funtu.

• CDS serija je identična sa CD, uz dve male razlike: disk jedinice imaju prekidač pomoću koga se bira mod sa 40 ili 80 traka i, naravno, koštaju 469 za CD400/S, odnosno 499 funti za CD800/8. Ovi kapaciteti, razumljivo, mogu da se udvostruče kupovinom „double density" disk interfejsa.

. . . i diskete u izlogu

I nabavka disketa može da bude svojevrstan problem. Na tržištu se, naime, nudi zaista mnogo tipova. Najvažnije je da disketa koju kupujete bude primerena vašoj disk jedinici, mada i od toga, kao što ćemo videti, može da bude izuzetaka. Ukoliko posedujete disk jedinicu sa 80 traka, treba da kupite pakovanje diskete na kojima je napisano „80 track". Ukoliko vaš uređaj ima dve glave, na disketama treba da piše i „double sided". Ukoliko, najzad, posedujete „double density" disk interfejs, i ovaj napis treba da pročitate na kutiji disketa. Varijacije u ceni su značajne — 10 S/S D/D („single sided — double density") disketa košta 17, a isti broj D/S D/D disketa celih 28 funti!

Neki vlasnici disk jedinica sa jednom glavom kupuju „dvostrane" diskete, računajući da će jednoga dana prodati svoju disk jedinicu i kupiti skuplju i bolju, zašto bi tada bacali sve diskete? Ovi vlasnici imaju još jedan razlog da smatraju da su učinili dobar posao kupujući „dvostranu" disketu — dovoljno je da je izvade iz drajva, okrenu za 180 stepeni i ponovo vrate na mesto, pa će računar pisati i po drugoj strani diskete, čime smo za 28 funti dobili jednak skladišni prostor kao da smo kupili 20 S/S disketa, za koje bismo platili 6 funti više!

Treba, međutim, reći da se sve diskete prave tako da budu D/S D/D. Posle toga svaka disketa prolazi kroz veoma strogu kontrolu kvaliteta — određen broj u rastur, dok se preostale obeleže kao S/S S/D ili S/S D/D. Testovi koji su pre toga izvršeni su, rekosmo, izuzetno strogi, tako da 90% disketa koje su označene kao „jednostrane" mogu sasvim lepo da se koriste i kao „dvostrane". Autor ovoga teksta je napravio jedan eksperiment ovoga tipa — kupio je jednu kutiju od 10 S/S S/D disketa firme Control Data i pokušao da ih formatira kao D/S D/D. Samo jedna strana jedne diskete je imala feleričnu traku, dok su ostale pokazale kao perfektne.

Oni koji poseduju „jednostranu" disk jedinicu, a zatim kupe „jednostrane" diskete, naići će na jedan problem kada pokušaju da iskoriste njihovu drugu stranu. Na svakoj S/S disketi nalazi se, naime, jedan mali otvor na desnoj strani centralnog otvora. Ovaj otvor omogućava pozicioniranje magnetne glave koja u početnom trenutku treba da nađe nulti trag („soft sectoring"). Okrenite disketu naopako i ništa neće raditi! Rešenje ovoga problema postoji — treba izbušiti simetričnu rupu sa leve strane (desne kada se disketa prevrne) centralnog otvora. Ne preporučujemo vam da ovo radite ukoliko niste već nekoga videli „na delu" (možda ćemo o ovoj operaciji pisati više u „Računarima 4"), a naročito ako su na disketi bilo kakvi važni podaci — posle nestručno izvedene operacije računar, možda, neće biti u stanju da ih pročita!

Jedno od pitanja sa kojima se suočava novopečeni vlasnik disk jedinice je i „koliko disketa da kupim?" Odgovor zavisi, najpre, od toga da li na svaku disketu treba da stane 100, 200 ili 400 Kb podataka, a zatim i od toga da li želite da se bavite nekim većim poslom ili, jednostavno, „kompletirate sistem". Čini nam se da je 10 ili 20 disketa sasvim dovoljno za početak, docnije ih možete dokupiti još. Ne zaboravite da podatke koje ste obradili i koji će vam možda zatrebati „jednoga dana" uvek možete da presnimite na kod nas lako nabavljive kasete! Na jednu kasetu C60, pri brzini snimanja od 1500 bauda, na svaku stranu staje oko 320 Kb podataka.'

u sledećem broju

VEZA SA DISKOVIMA
disk jedinice
i interfejsi za

- *BBC B*
- *komodor 64*
- *spektrum*

KNJIGE ZA VAŠ USPEH U ŽIVOTU...

BIBLIOTEKA SAZNANJA

1. **A.M. Semorie: „I VI MOŽETE IMATI SUPER PAMĆENJE"**

   — VI izd. Programirani priručnik tehnike pamćenja i učenje, kojim možete višestruko povećati svoje pamćenje i sposobnost učenja.

   I knjiga: **„METODE I TEHNIKE PAMĆENJA"** (202 strane) Din 600 -
   II knjiga: **„PRAKTIČNA PRIMENA"** (228 str.) — **Din.** 600 -
   PRAKTIČNE, LAKO SHVATLJIVE I SVIMA PRISTUPAČNE METODE **BRZOG, LAKOG I TRAJNOG UČENJA I PAMĆENJA svih vrsta podataka i znanja.**

   Komplet (obe knjige zajedno) Din. 1.100.-
   Format 17×24 cm, 430 str., preko 150 ilustracija, 15 samotestova.

2. **S. Yesudian — E. Haich: „JOGA I SPORT"**

   — Kompletan, praktičan i veoma pristupačan priručnik HATHA JOGE, drevne i **širom sveta potvrđene** jedinstvene umetnosti Istoka za **sticanje i očuvanje dobrog zdravlja tokom celog života.**

   Ko želi **DOBRO ZDRAVLJE I LEPU LINIJU, PSIHIČKI MIR I SNAGU** treba da prouči ovu veoma **rasprostranjenu knjigu prevedenu na 19 jezika sveta** u preko dva miliona primeraka.

   Cena Din. 600.- džepni format, 224 str., preko 70 ilustracija.

3. **S. Yesudian: „JOGA I ZDRAVLJE"**

   — Viši kurs HATHA JOGE, nastavak knjige „„JOGA I SPORT".

   Programirani priručnik primene najboljih metoda Hatha joge za svaku sedmicu u godini.

   SPECIJALNI REGISTAR I DIJAGNOSTIČKI INDEKS za preko 140 oboljenja omogućuju **efikasnu primenu najboljih metoda za pojedine organe i oboljenja. Cena Din.** 600.-, džepni format, 221 str., preko 250 ilustracija.

   Komplet (obe knjige „JOGA I SPORT" I „JOGA I ZDRAVLJE" zajedno) Din. 1.100.-

Sve knjige su štampane latinicom, na finom ofset papiru sa koricama u više boja.

GARANCIJA: Svakom nezadovoljnom čitaocu naših knjiga odmah vraćamo novac, ako neoštećenu knjigu vrati u roku oc pet dana nakon prijema.

Kod isporuka za inostranstvo zaračunavamo poštarinu i pakovanje.

BIBLIOTEKA SAZNANJA

NARUDŽBENICA — RAČ. III

1. **SUPER PAMĆENJE** _____ **kom.**
2. **JOGA I SPORT** _____ **kom.**
3. **JOGA I ZDRAVLJE** _____ **kom.**

Adresa naručioca (ispuniti čitko, ŠTAMPANIM SLOVIMA)

Ime i prezime _____

Ulica i broj _____

Pošt. br. i mesto _____

Knjige plaćate pouzećem — poštaru kod preuzimanja

Naručite na adresu: „BIBLIOTEKA SAZNANJA"
p.p. — 20/70, 11031 BEOGRAD 8.

OVOG MESECA U PRODAJI

prof. dr Nedeljko Parezanović

BEJZIK ZA GALAKSIJU

u izdanju časopisa „Galaksija" i Zavoda za udžbenike i nastavna sredstva

---

KNJIGA ZA SVE VLASNIKE RAČUNARA „GALAKSIJA"

Mikroračunar „Galaksija" danas se nalazi u posedu sve većeg broja pojedinaca, škola i drugih ustanova. Ova knjiga predstavlja svima njima neophodno štivo — priručnik za programiranje na bejzik jeziku za ovaj računar.

BEJZIK ZA GALAKSIJU

sastoji se iz tri glavne celine:

UVOD

upoznaje čitaoca sa osnovnom konfiguracijom računarskog sistema i bejzik interpretatorom ugrađenim u njegov ROM:

- *struktura*
- *komande*
- *tastaturne komande*
- *izveštaji o greškama*

PROGRAMSKI JEZIK BEJZIK

govori o specifičnoj implementaciji ovog programskog jezika primenjenog u „galaksiji":

- *azbuka*
- *osnovne sintaksne jedinice*
- *linijski programi*
- *razgranati programi*
- *programi sa ciklusima*
- *potprogrami*
- *rad sa datotekama*
- *ekran i grafika*
- *BASIC-mašinski jezik*
- *programske tehnike*

DODATAK

obuhvata strogi opis programskog jezika i komandi interpretatora:

- *opis programskog jezika*
- *komande interpretatora*
- *tastaturne komande*

sintaksa i semantika bejzik jezika

BEJZIK ZA GALAKSIJU

je korisno štivo za svakoga ko želi da uđe u tajne profesionalnog programiranja, priručnik za upotrebu računara, udžbenik

160 strana, format 17×24 cm, četvorobojne plastificirane korice, povez spirala

cena 700,00 dinara

NIJEDAN VLASNIK RAČUNARA „GALAKSIJA"
*BEZ KNJIGE **BEJZIK ZA GALAKSIJU***

računar je muškog roda

Računari iz drugog ugla

O čemu razmišlja saradnica „Računara" dok sluša pesmu „Program tvog kompjutera"

Crtež: Elizabeta Novak

To je jasno i onima koji nemaju baš mnogo veze sa gramatikom, ali se bar malo razumeju u anatomiju. Računar je, baš kao i druge muške naprave (automobili, avioni . . .) opremljen džojstikom koji muškarcu pruža vrhunski doživljaj sigurnosti, moći i samopotvrđivanja. Sirote devojčice nemaju za šta da se uhvate. Muškarci imaju monopol i nad brojevima, a neki još uvek misle da za ženu najveće dostignuće u toj oblasti predstavlja izvlačenje dobitnih kombinacija za Loto, utorkom uveče, na prvom programu. Biti žena — to je, ako ne nedostatak, a ono barem čudno svojstvo za nekog ko se bavi računarima.

Karte su odavno podeljene; još od detinjstva se zna šta je za koga: dečki se zabavljaju puškama, pištoljima, i uopšte, agresivnim, muškim stvarčicama, dok se nesretne devojčice igraju kuvanja ručka i trude se da budu dobre mame svojim lutkama.

Ali, došlo je novo vreme. Dečki više nemaju vremena za devojčice. Suviše su zaokupljeni svojim računarom. Interesantno je da se zaraza video igrama, baš kao i sida, prenosi isključivo među muškarcima. Uostalom, sa računarom je sve mnogo lakše: jednostavniji je za rukovanje, ne zahteva ništa, ne prenemaže se, i na kraju, može i da se isključi. Malo je devojaka koje raspolažu tim kvalitetima. Zato one mogu da budu samo program njegovog kompjutera.

A zašto tom čudesnom napravom zvanom računar ne bi upravljala ženska ruka?

Ima raznih žena. „Ona" može biti anđeo ili žena-cvetić, pasivno biće koje čeka da njega sleti pčelica. Cilj takve žene je da se dopadne po svaku cenu. Ako to moda zahteva, i ona će se uključiti u „čip-revoluciju". Za nju će biti važno da je tastatura u istom tonu kao i njena haljina, a boja displeja da se slaže sa bojom njenih očiju.

Živ čovek svašta može, a tek živa žena! Nije čudno kada se poneka od njih „prešaltuje" ne neke tipično muške oblasti. Ako im bavljenje tim „neženstvenim" zanimanjima još i lepo stoji, na to se gleda s blagonaklonošću, jer, između ostalog, time se umiruje sopstvena savest, pokazuje dobra volja, a takvi pojedinačni slučajevi nikog ne ugrožavaju.

Feministkinja je ekstremna. Žena-borac. Ona se u svemu nadmeće s muškarcima. Od nje bi se moglo očekivati da se sadistički iživljava na sirotim malim mikroračunarima, da im uvrće uši ne bi li time nešto dokazala.

Ali, i feministkinje se udaju; domaćica je sudbina svake žene — znači, domaćica je najreprezentativniji primerak ženskog roda. Muževi domaćice zaključavaju u kuhinju — to je njena teritorija na kojoj ona može da utiče na stvari i u okviru koje pronalazi sumnjivu sigurnost.

Međutim, žena uzalud zatvara vrata i zastire prozore — ona ne nalazi apsolutnu sigurnost ni u svom domu; i njega opseda taj muški svet, koji ona izdaleka poštuje, ne usuđujući se da se upusti u njega.

Dolazi novo vreme. Vreme računara.

To je bolan trenutak za do[illegible]e koje znaju da koriste samo dva od dvadeset mogućih programa mašine za pranje rublja.

Računar kućni prijatelj? Žena nema baš nikakve koristi od takvog prijatelja. Muž je vara sa tim kućnim prijateljem na njene oči. Međutim, žene se svakako neće pomiriti sa takvom sudbinom. Domaćica je već uspela da preživi prethodnu tehnološku revoluciju, kada je u kuhinje uvedena struja. Ona će pripitomiti i računar. Možda bi neka od njih mogla da postane pravi virtuoz u tome, pa će te jednog dana gledati film o domaćici koja je zloupotrebila svoju mikroprocesorsku pećnicu i umalo izazvala treći svetski rat! Da vam se kosa digne na glavi. Ili zamislite: domaćica uz pomoć svoje hiperpotentne multivalentne procesorske haube kidnapuje avione koji su zašli u njen vazdušni prostor!

Ali, takve stvari ostaju rezervisane za savremene filmske bajke namenjene deci i omladini. To što klinci beže iz škole u salone zabave gde im otimaju i pare i život, ne smatra se opasnim. Sve je to već viđeno, samo u drugom obliku, kažu pedagozi. Zaraza video igrama će proći, pojaviće se neka nova moda. Čak i kada ne bi bilo drugih argumenata, to mora da je istina, jer oni to tvrde s toliko odmerenosti i mirnoće u glasu.

A šta bi se dogodilo da računar izvrši svoj destruktivni udar u ženu, u bitnu kariku porodice, koja je bitna karika društva?

Ženama je logika oduvek bila slaba strana; pomoću jednog silogizma ne može se napraviti dobra ruska salata, a ni utišati plač deteta. Ali, moglo bi se desiti da su žena i računar idealan par: ženska intuicija+računarska logika — za muškarce zastrašujuća kombinacija. Ako je tako nešto moguće, nije poželjno. Život bi dobio previše ljut začin. Ne, muškarci to ne smeju dozvoliti.

(Muške) sede glave koje paze na dobrobit ovoga sveta, oni kojima je u interesu da nastave sadašnjost, liju gorke suze nad prošlošću koja će uskoro iščeznuti. I kosa im se diže na glavi, ukoliko je još imaju, pri pomisli na budućnost. Oni kuju planove: žena će se kad-tad sresti sa računarom; samo da to ne bude u proleće, da se ne desi ljubav na prvi pogled! Najbolje je da se sretnu na njenoj teritoriji, u kuhinji (romantika uz krpu za sudove). Računar treba staviti u peglu, u viklere za kosu i ostale naprave namenjene mučenju ženskog roda. Ne ostavljajte je nikad samu uz računar dok neko svira! Bliske susrete žene i računara treba strogo kontrolisati.

A žene ćute, ne bune se. Trude se da budu zadovoljne malim stvarima, kad im već velike nisu pri ruci. Uostalom, šta tu ima mnogo da se priča; za obračun kućnog budžeta dovoljan je i „digitron".

Jelena Rupnik

program od milion dolara

Softver na tekućoj traci

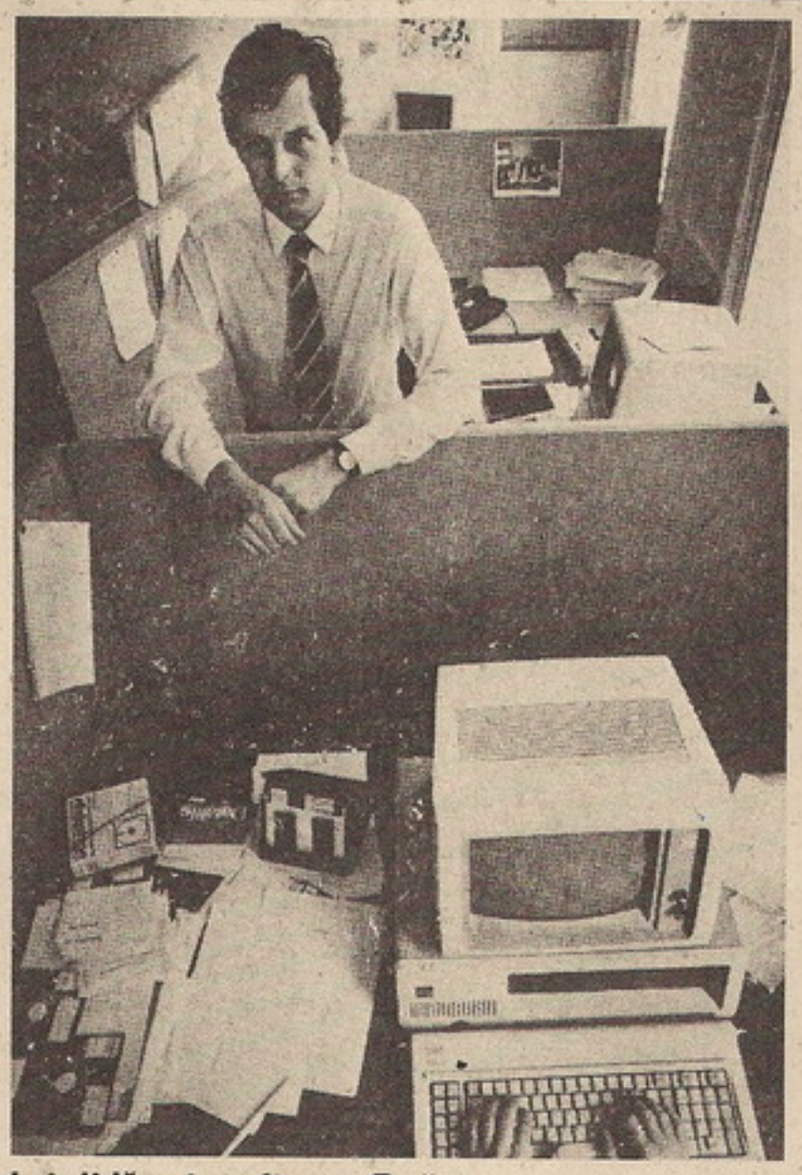
Skup uređaj, razume se, ne obezbeđuje sam po sebi uspešne programe — neki amateri su imali sreće u razvoju softvera na „spektrumima" kod kuće. Uprkos tome, neka izrazito talentovana deca postala su pravi majstori učestvujući u razvoju programa za velike softverske kuće u nekoliko poslednjih godina. Moćni računari ovih kuća i sredstva za programiranje visokog nivoa omogućuju im veliku prednost u odnosu na vlasnike kućnih računara, kao i produktivnije programiranje.

Gumeni dugmići . . .

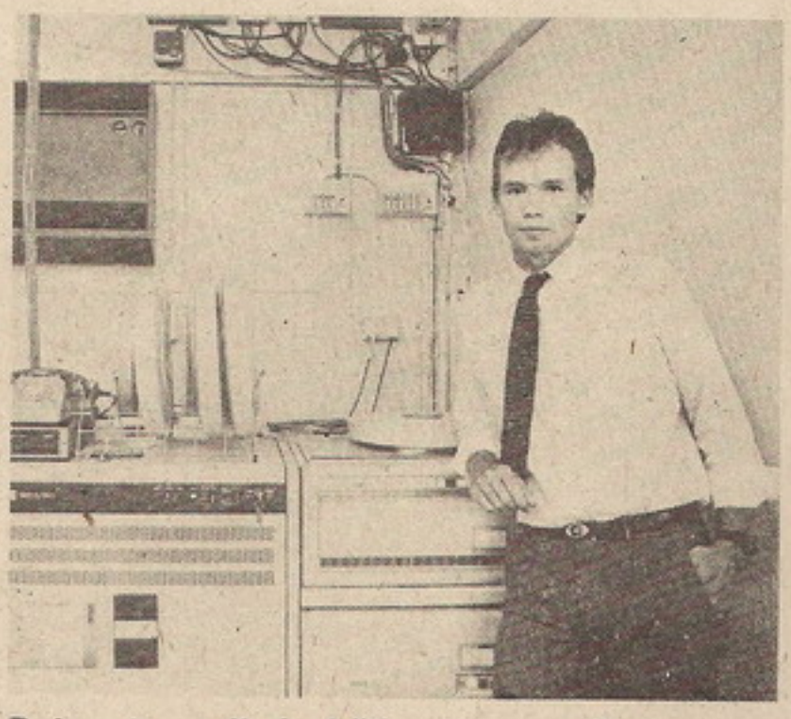
Jedna od najvažnijih karakteristika ozbiljnog softvera za kućne računare je brzina rada; ovo znači da programi moraju (barem delimično) da budu pisani u mašinskom kodu. Ali, rad sa mašinskim kodom je prilično težak — programerima koji rade u mašinskom kodu potreban je razvojni softver koji će im pomoći u izradi njihovih programa. U najboljem slučaju, kod asemblerskih programa, kôd izvornog programa se mora prevesti u objektni kôd, koji mašina raspoznaje, a to je popriličan posao ako se radi o velikom programu. Mnogi pisci softvera rade na ovaj način. Da bi napisali program za „spektrum", na primer, oni koriste asemblerski program koji radi na istoj mašini. Ovakav metod ima svoja ograničenja.

Prvo, kvalitet asemblerskih programa za rad na kućnim računarima je loš. Čak i najjednostavniji od ovih paketa uzima znatan deo memorije i, samim tim, smanjuje veličinu programa koji se mogu pisati pomoću njega. Mnogi kućni računari su, takođe, neprikladni za rad na duže vreme: slaba tastatura, slab prikaz na ekranu, i, najčešće, nedostatak disk jedinice mogu uticati na to da rad sa takvim uređajem bude neprijatan.

Iz ovih razloga većina profesionalnih kompanija ne koristi mirkoračunare za razvoj programa koji su, inače, njima namenjeni (ove mikroračunare zovu „ciljna mašina"), već poslovne računare, koji raspolažu posebnim softverom (zovu ih „sistemi za razvoj"). Programeri koji rade na ovim mašinama često koriste jezike kao što su paskal i C, posebno verzije ovih jezika koji su poznati kao kompajleri sa jezika na jezik (cross-compilers ili cross-assemblers) koji omogućavaju da se radi, na primer, na mikroračunaru sa procesorom 8086, dok će programi raditi na mašini sa procesorom Z80. Ovi prevodioci sa jezika na jesu, u stvari, jezici visokog nivoa, kao bejzik, što olakšava rad programerima, ali su programi koji se pomoću njih prave napisani u mašinskom kodu. Programeri koji dobro rade u mašinskom kodu kontrolišu programe koji se razvijaju, a često rade i na njihovoj optimizaciji.



Intelidžent softver: Pošto se specijalizovao za strategijske igre, kao što je šah, IS za razvoj softvera koristi IBM i Apple računare, uz dodatak interfejsa koji je sam razvio; podela programa na opšte segmente i one koji zavise od mašine olakšava IS-u da pruža programsku podršku čitavom nizu računara i mašina nemenjenih za igranje šaha



Psion to radi drukčije: Kompanija je 1982. godine kupila dva „vaksa 750" kao osnovu za razvoj svog softvera; na svakoj mašini može istovremeno da radi do 20 progremera; svima na raspolaganju stoje razni prevodioci sa jednog jezika na drugi, softverske biblioteke i programi za ispravljanje grešaka prilikom kreiranja i prevođenja programa.

. . . i sistemi za razvoj

Svakako da ovakav sistem za razvoj ima velike prednosti u odnosu na kućni računar. Asembler orijentisan na diskove, ili onaj koji koristi poluprovodničku memoriju za memorisanje velikih tabela, radiće mnogo efikasnije od asemblera skinutog sa trake i koji je ograničen mogućnostima kućnog mikroračunara. U razvojnu verziju koda mogu se ugraditi rutine za „čišćenje" grešaka, bez bojazni da će kod biti prevelik za memoriju. Na poslovnom računaru je bolje raditi i zato što ima dobru tastaturu, jasan prikaz na ekranu i disk jedinice.

Firma koja koristi ovu tehniku razvoja softvera je Intelidžent softver (Intelligent Software — IS); osnovali su je 1981. godine Dejvid Livaj (David Levy) i ANT Mikrover (ANT Microware) Roberta Medžija (Robert Madge). Kompanija se specijalizovala za strategijske igre, uglavnom pisane po narudžbini za razne kućne računare, kao i za softverski deo posla za mašine namenjene za igranje šaha. Mada u igrama kao što su šah ili bridž nema spektakularnih borbi kao u drugim igrama, mnogo toga se odvija iza ekrana. Kao i ostale igre, i strategijske igre zahtevaju brzinu koju može da obezbedi jedino asembler.

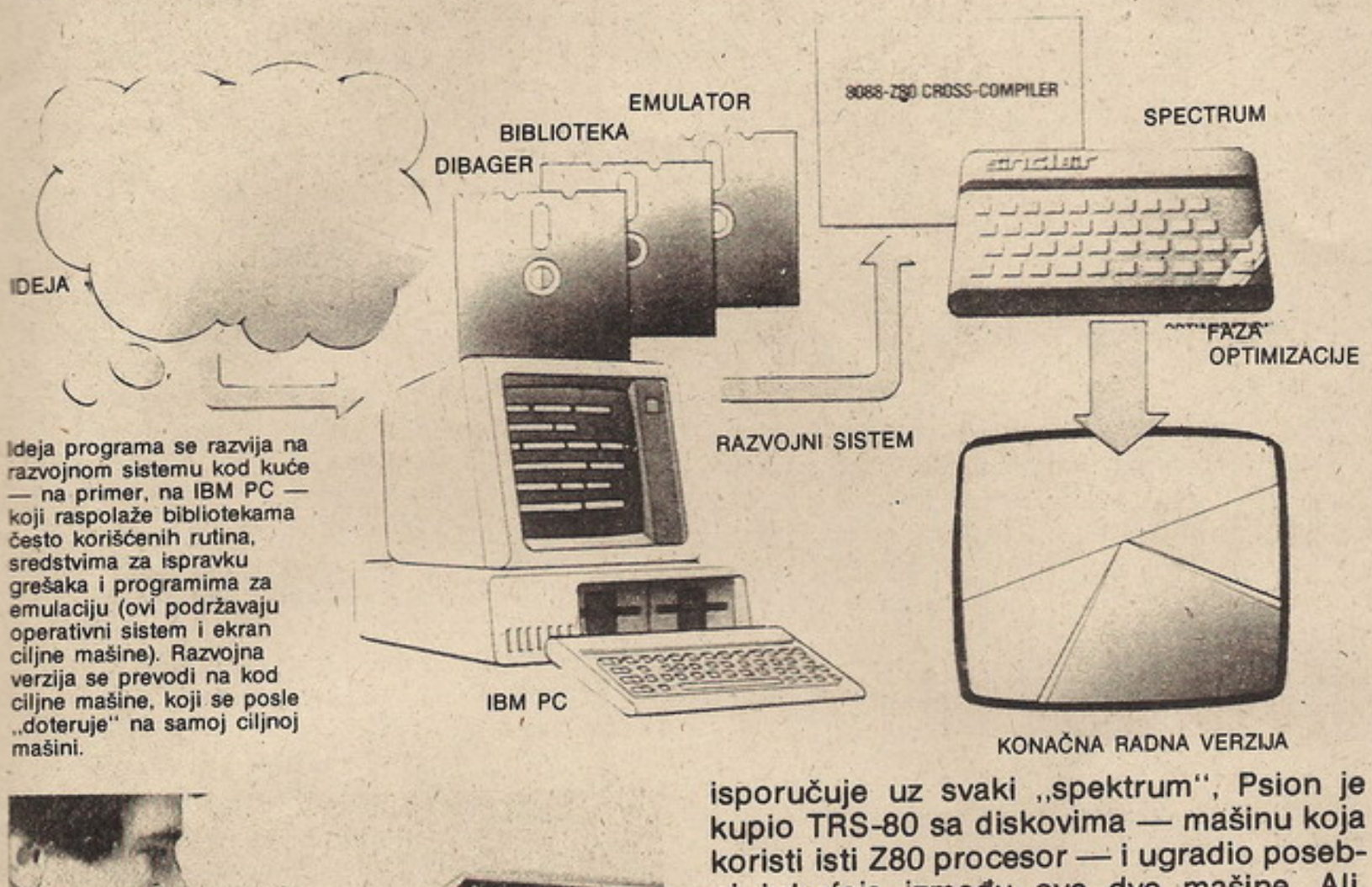
Pored toga što koristi samu ciljnu mašinu za razvoj softvera, IS koristi i računare kao što su IBM PC i „epl", uz obavezan interfejs koji omogućava razmenu koda između različitih mašina. Od kompanije se često traži da uradi konverziju nekih projekata — na primer, prebacivanje igre šaha sa jednog računara na drugi — tako da su programeri ove firme naučili da pišu kôd u obliku koji se lako segmentira. Jedan nivo segmentacije koji potvrđuje korisnost postupka kad je vreme u pitanju je podela programa na kôd koji se tiče same igre i kôd koji rešava pitanje ulaz/izlaz. Ulazno/izlazni kod će na novoj mašini imati drugu adresu u memoriji, a možda će se razlikovati i sa stanovišta strategije. Ovde je, maltene, potrebna ingenioznost da bi se podvalilo svim ograničenjima koja uslovljava hardver, ali praktično ne postoji zabranjeni kvantitet ulaza/izlaza. Kôd koji se tiče same igre ima daleko više mogućnosti za realizaciju, a pošto je izolovan od hardvera (izuzev procesora), njegova konverzija će biti jednostavna.

Tamo gde programeri rade kod kuće za softverske kompanije i gde svako razvija svoj softverski projekat, malo se ulaže u resurse. U ovakvim slučajevima individualnost je sačuvana po cenu velikim delom duplog posla, jer svaki programer ponovo otkriva kod za slične rutine.

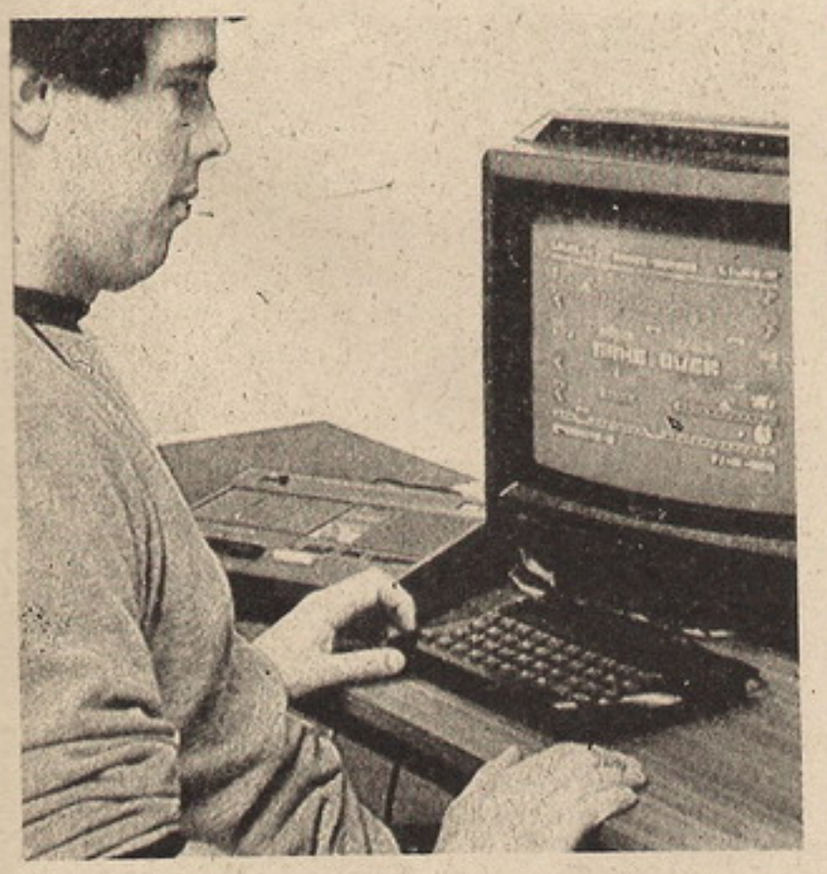
Jedna druga softverska kuća, Psion, koristi računare većih mogućnosti od IBM PC. Među britanskim softverskim kućama koje proizvode za isto tržište igara za kućne računare, Psion je jedini koji gro svog

Nema iole zagriženijeg ljubitelja kućnih računara koji ne sanja da napiše programski bestseler. Malo je, međutim, koji amater tu svestan sa čim se treba uhvatiti u koštac. Iza profesionalnih softverskih kompanija stoje enormni resursi pomoću kojih se prave programi vrhunskog kvaliteta. Jedna britanska softverska kuća ima poslovni sistem u vrednosti od 250 000 funti čija je namena isključivo izrada programskih paketa za kućne računare. Ovim tekstom, svakako, ne želimo da potkrešemo krila domaćim hakerima koji pate od slične slatke nesanice.

REALIZACIJA ZAMISLI



Ideja programa se razvija na razvojnom sistemu kod kuće — na primer, na IBM PC — koji raspolaže bibliotekama često korišćenih rutina, sredstvima za ispravku grešaka i programima za emulaciju (ovi podržavaju operativni sistem i ekran ciljne mašine). Razvojna verzija se prevodi na kod ciljne mašine, koji se posle „doteruje" na samoj ciljnoj mašini.



Vizuelni efekti: Programeri koji rade kod kuće na ciljnim mašinama mogu svojim programima da ostvare fantastične vizuelne efekte; kada odluče koju će koncepciju i pravila imati igra, razvijaju posebne rutine u običnom asembleru (Z80 ili 6502); na primer, koriste HiSoft „devpak" na „spektrumu"

razvoja izvodi na miniračunarima. Samo hardverska instalacija ove kuće vredi četvrt miliona funti.

Dva-tri „vaksa" . . .

Psion je počeo sa razvojem softvera za ZX81 i za to je koristio baš ZX81. Kada je krenuo na kreiranje „Horizon" trake koja se isporučuje uz svaki „spektrum", Psion je kupio TRS-80 sa diskovima — mašinu koja koristi isti Z80 procesor — i ugradio posebni interfejs između ove dve mašine. Ali, avgusta 1984. godine kompanija dolazi do zaključka da joj se ne isplati da menja ceo razvojni sistem svaki put kad se novi kućni računar pojavi na tržištu. Tako je stečeni profit investirala u kupovinu jačeg hardvera sa daleko većim brzinama rada. U principu, ovaj bi hardver mogao biti dovoljno fleksibilan da može da savlada sve probleme koji će se javiti sa pojavom novih računara u budućnosti. Izabrane su dve mašine „vaks 750" koje rade pod DEC-ovim operativnim sistemom VMS.

Ove dve mašine omogućile su Psionu dve prednosti: kvalitet softvera koji ima DEC, sa mogućnošću kreiranja posebnih softverskih pomoćnih sredstava, i moćnu kombinaciju operativnog sistema i hardvera. Ima dovoljno mesta za smeštanje pomoćnih softverskih sredstava, kao što su kompajleri, biblioteke opštih potprograma i programa za otklanjanje grešaka; sve ovo može da deli 16 do 20 programera koji istovremeno rade na istoj mašini. Dve mašine omogućuju da se softver, kada je to potrebno, lako prenosi sa jednog na drugi kompjuter.

Biblioteke sa opštim potprogramima već su bile deo Psionove filosofije u danima TRS-80, ali prebacivanje sa jednog na drugi disk nije bilo jednostavno. Nova „oaxova" mašina omogućava timovima programera da rade zajedno, da dele zajedničke biblioteke, iz kojih se moduli mogu pozivati u svako vreme; biblioteke mogu deliti i timovi koji rade na različitim projektima. Ovo je velika prednost sistema koji radi u raspodeljenom vremenu (timesharing system); pored toga, on oslobađa programere onog administrativnog dela posla, jer na sistemu ostaje zapisano ko je koliko radio. Psion, čak, namerava da kupi treći „vaks" isključivo za administrativne poslove.

Daleko je Psion

Čak i kad biste sebi mogli i da dozvolite kupovinu jednog „vaksa", ne biste stali rame uz rame sa Psionom. DEC je Psionu samo mnogo omogućio, a malo dao. Psion je ugradio u sistem mnogo svojih pomoćnih softverskih sredstava.

Psion koristi jezik C, „posredničkog nivoa" (intermediate-level language), koji proizvodi prilično kompaktan i brz objektni kod za šesnaestobitne čipove, kao što je 8086, mada je daleko od osmobitnih C kompajlera. Da bi napravio softver za ciljnu mašinu tipa „spektrum", Psion je morao da razvije posebne tehnike rada. Psion nije voljan da otkriva svoje tajne, ali se zna da je kompanija koristila jezik C za pravljenje svog kompajlera, koji je nazvala „naš jezik tabele". On liči na jezik C, može se prenositi sa jednog procesora na drugi, a izbacuje zavidno efikasan kôd.

Opšte je pravilo da na održavanje sistema i izradu vlastitog softvera, kao što su tabelarni jezici, odlazi 30 odsto programerskog rada, ali Psion smatra da mu se može dati i više. Kad izvorni kod razvijate u svojoj kući, vi ste totalni vlasnik: možete uzimati posebne delove, poboljšavati ih ili prilagođavati ih na način na koji to nikako ne možete sa komercijalnim softverom.

Psionov softver uključuje i programe koji mogu simulirati popularne mikroprocesore, kao što su Z80 i 6502. Tako se ogromni „vaks" računari mogu ponašati kao da su „komodor 64" ili „spektrum". Bez obzira na brzinu rada „vaks" računara, simulatori rade brzinom kojom rade ciljne mašine. Programer ima tu prednost da prati sadržaje svakog registra u mikroprocesoru u bilo kojoj fazi rada programa. Ovo je posebno važno za otkrivanje grešaka u programu. Prirodno, kad program, pisan u mašinskom kodu, ima grešku i kad stane sa radom, programer ne može da kaže šta je bilo pogrešno. Psion je tako uštedeo mnoge časove koji bi otišli na ispravljanje grešaka u programu.

Mnogo od Psionove razvojne koncepcije je uključeno u izradu četiri poslovna programa za Sinclair QL. Familija Motorola 68000 čipova, od kojih se jedan ugrađuje u QL, koristi više programske jezike, a C programi kompiliraju tako dobro na ovim čipovima da pisanje u asambleru postaje suvišno. Ako bi svi kućni računari sledili QL, jezik C bi mogao u potpunosti da zameni asembler, a Psion i male softverske kuće bi zauvek mogle da zaborave ručno kodirane translatore.

neki bolji bejzici

Programi koje treba imati

BETA BASIC

O beta bejziku, jednom od najmoćnijih programskih proširenja računara „spektrum", pisali smo u avgustovskom broju „Galaksije". Program je, međutim, dovoljno vredan da zaslužuje punu pažnju i u našem specijalnom izdanju. Govorićemo, zapravo, o najnovijoj verziji beta bejzika, kojoj su njeni autori nadenuli oznaku „1.8". Iza te oznake stoji oko devet kilobajta mašinskih rutina, koje dopunjavaju bejzik „spektruma" do zaista impozantnih razmera — programeru stoji na raspolaganju preko 50 novih naredbi i funkcija, uz mnoga poboljšanja već postojećih mogućnosti računara.

Možda sumnjičavo vrtite glavom jer biste radije zadržali svojih 9K za sebe nego što biste ih ustupili proširenju bejzika. Znamo da je za sastavljanje ma kako složenih programa dovoljan i mali broj naredbi. Obogaćivanje nekog jezika (ako baš hoćete, isto važi i za govorni jezik) nema, međutim, za cilj da pruži **suštinski** nove mogućnosti u izražavanju ideja, već da olakša to izražavanje, uštedi vreme i, sasvim sigurno, memoriju. Tih 9K koje ćete poveriti beta bejziku ionako biste brzo ispunili koristeći „spektrumove" ne baš tako moćne naredbe. U njemu su, za ono za šta je dovoljna jedna reč, ponekad potrebne rečenice.

Naredba beta bejzika SORT, na primer, vrši uređenje numeričkog ili alfanumeričkog niza bilo u rastućem, bilo u opadajućem poretku, po želji. Primena te jedne jedine naredbe oslobađa vas pisanja odgovarajućeg potprograma, koji, naravno, uvek možete sastaviti, ali nikada nećete postići isti efekat. Jer, ako rešenje ostvarite u bejziku, trebaće vam beskrajno strpljenje dok dočekate da vam takav program obavi uređenje nekog niza sa većim brojem elemenata. A izlet u mašinski jezik nije uvek naročito privlačan. Uostalom, čak i tada, u najboljem slučaju, rešenje će biti uporedivo sa onim koje koristi naredba SORT. Da li se trud onda uopšte isplati?

Ipak, najjači argument u korist proširenja programskog jezika jeste mogućnost sastavljanja programa koji su daleko pregledniji i privlačniji sa estetske strane. U programiranju se uvek teži ka što boljoj **struktuiranosti** programa. Program treba da bude pregledan i lako razumljiv za ljude, koji ga čitaju. Računar će, doduše, uvek sa njim izaći na kraj, ali programer teško. Sigurno ste pokušavali da anlizirate programe koje niste vi pisali, ili, ako i jeste, to je bilo davno. Priznaćete da je to sve samo ne lak posao: gomila GO TO naredbi koje jure jedna drugu, zamršene višestruke petlje kojima ne sagledavate ni početak ni kraj . . . A program se **može** pisati i tako da je jedan pogled na njega dovoljan za razumevanje njegovog rada. istina, programski jezik bejzik nije baš stvoren za struktuirane programe, ali sa beta bejzikom se može dosta toga postići.

Usavršen editor

Šta konkretno nudi beta bejzik? Pre svega, postoji velika grupa naredbi koja olakšava unošenje programa, jer su uvedena mnoga poboljšanja u „spektrumov" editor (editor je mašinska rutina koja kontroliše kreiranje bejzik programa). Pored nekoliko manje značajnih novina (trepćući programski kursor i generisanje tona pri pritisku svakog tastera), poboljšanja obuhvataju i nekoliko naredbi bez kojih je rad sa računarom prosto smešan! Mislimo, pre svega, na naredbu EDIT, koja omogućava trenutni pristup bilo kojoj programskoj liniji, bez obzira gde se u tom trenutku nalazi programski kursor. Osim toga, naredbi LIST je omogućeno da izlista samo deo programa, ili, čak, samo jednu liniju. Postoji i naredba AUTO za automatsko generisanje linijskih brojeva sa zadatim korakom. Naredba DELETE će prosto izbaciti iz programa neki blok koji vam više ne treba. Naredbom JOIN možete spojiti dve linije u jednu, a naredbom SPLIT razdvojiti jednu liniju na dve. Konačno, možete i prenumerisati vaš program, ili samo deo programa, pod kontrolom naredbe RENUM.

Interesantna je i mogućnost da svakom tasteru dodelite neki niz znakova, koji će se kasnije ispisati čim pritisnete taj taster. Naredba koja to omogućuje je DEF KEY. Možete, čak, tasteru pripisati i niz naredbi, kao mali program sa linijskom strukturom. Kada kasnije pritisnete taj taster, izvršiće se odgovarajući program — izuzetno praktično za pozivanje mašinskih potprograma. Jedna od najmoćnijih naredbi beta bejzika je KEYIN, koja deluje na proizvoljan niz karaktera naveden iza naredbe, a rezultat je isti kao da ste pod kontrolom editora otkucali taj isti niz i zatim pritisnuli ENTER. Zamislite, recimo, neki program koji ima i jednu liniju oblika 150 KEYIN „10 PRINT x". Kada dođe do linije 150, program će, prosto, **sam sebi dopisati** liniju 10 PRINT x, a onda nastaviti dalje izvršenje. Naredba omogućuje pisanje programa koji razvijaju sami sebe! Ne mogu se ni sagledati mogućnosti koje odatle proističu.

Beta bejzik ima i naredbu TRACE koja olakšava testiranje programa i otkrivanje grešaka. Neposredno pre izvršavanja svake naredbe, izvršiće se skok na potprogram koji želite. U okviru tog potprograma možete štampati linijski broj i broj naredbe koja će uslediti, možete izlistati tu liniju ili varijable koje vas interesuju itd.

Sistemske naredbe

Drugu grupu naredbi mogli bismo uslovno nazvati grupom „sistemskih" naredbi, jer one omogućuju vezu sa sistemskim rutinama računara (npr. sa bejzik interpretatorom) i olakšavaju pristup memoriji. Pre svega, tu treba pomenuti naredbu ON ERROR, pomoću koje je moguće programski rešavati sve probleme koji se jave u toku izvršavanja. Kada god dođe do neke greške, onda će umesto uobičajenog prekida i standardne poruke doći do skoka na neki zadati potprogram, u okviru koga možemo promeniti prirodu nastale greške i, u skladu sa tim, nešto preduzeti. Na primer, ako crtamo grafik neke nepoznate funkcije pa se desi da u jednoj tački funkcija nije definisana, ili je veoma velika tako da tačka ispada van ekrana, možemo prosto ignorisati tu tačku i preći na sledeću, a na kraju štampati obaveštenje o problemima koji su se uz put javili.

Beta bejzik olakšva rad i sa mašinskim programima, jer je omogućen jednostavan prekid većine mašinskih rutina prostom upotrebom komande BREAK (iste one kojom se prekida i bejzik program).

Pomenućemo i naredbu CLOCK pomoću koje „spektrum" dobija prilično precizan časovnik. Vreme se može ispisivati na ekranu, a moguće je i vremenski kontrolisati razne akcije koje računar preduzima (npr. izvršenje određenog potprograma u zadatom trenutku i slično).

Naredba POKE može se primenjivati i na alfanumeričke nizove, a rezultat je smeštanje svih karaktera niza u memoriju, redom počev od zadate adrese. Postoji i nova naredba DPOKE (skraćeno od Double POKE), koja smešta u memoriju 16-bitni broj, pri čemu se istovremeno pune dve lokacije. Ova naredba, zajedno sa odgovarajućom funkcijom DPEEK, znatno pojednostavljuje operisanje sa adresama; tu posebno mislimo na sistemske varijable.

Kontrola ekrana

Treća grupa naredbi se odnosi na kontrolu ekrana. Sigurno ste do sada više puta bili u situaciji da pravite programe koji nešto računaju, a zatim daju rezultate u obliku liste ili tabele sa nekoliko kolona. U vrhu ekrana obično planirate prigodno zaglavlje, na čije oblikovanje uložite dosta truda. A onda, kada listanje počne, i tabela popuni ceo ekran, u dnu se pojavi jedno lakonsko pitanje: „scroll?". Već znate da postoje dva rešenja u tom slučaju. Ako odgovorite sa NO, doći će do prekida celog programa, što uopšte niste želeli! A ako odgovorite sa YES, listanje će se nastaviti i vi ćete prisustvovati nepovratnom ispraćaju vašeg zaglavlja . . . U najboljem slučaju, zaglavlje će brzo nestati, ostavljajući samo podatke da se besmisleno nižu.

Sa beta bejzikom ovakvih problema više

Onog trenutka kada za njegov kompjuter na tržištu počnu da se pojavljuju komercijalni programi tipa „sto najboljih mašinskih rutina“ ili novi dijalekti bejzika, ljubitelj računara može slobodno da počne da razmišlja o novom modelu. To je najbolji znak da je ROM njegovog ljubimca šupalj kao švajcarski sir i da nikakva proširenja neće moći da naprave od njega dobru mašinu. Proširenja bejzika mogu, doduše značajno da povećaju snagu računara, ali nikada i da postanu njegov organski deo. Ovakvo proširenje, ako se ne nalazi u ROM-u na kartici, mora uvek iznova da se učitava u računar, bez obzira da li se program piše ili izvršava, program u proširenom bejzika može biti od koristi samo onima koji imaju isto proširenje i . . . komplikacijama nikad kraja. Postoje, međutim, dva programa koje se isplati imati stalno u računaru — „beta bejzik“ za „zx spectrum“ i „sajmons bejzik“ za „komodor 64“ — čak i onda kada ih treba svakodnevno učitavati sa trake.

REČNIK BETA BEJZIKA

ALTER	PROC	NUMBER
AUTO	RENUM	OR
BREAK	ROOL	RNDM
CLOCK	SCROLL	SRCN$
DEF KEY	SORT	SINE
DEF PROC	SPLIT	STRING$
DELETE	TRACE	TIME$
DO	UNTIL	USING$
DPOKE	USING	XOR
EDIT	WHILE	
ELSE	XOS	
END PROC	XRG	
EXIT IF	YOS	
FILL	YRG	
GET	AND	
JOIN	BIN$	
KEYIN	CHAR$	
KEYWORDS	COSE	
LIST	DEC	
LLIST	DPEEK	
LOOP	FILLED	
ON	HEX$	
ON ERROR	INSTRING	
PLOT string)	MEM	
POKE (string)	MEMORY$	
POP	MOD	

nikada nećete imati. Stoji vam na raspolaganju moćna naredba SCROLL za pomeranje slike gore, dole, levo ili desno, i to za proizvoljan broj pozicija (u visokoj rezoluciji!). Slično radi i naredba ROLL, s tom razlikom što se deo slike koji nestaje sa jedne strane ekrana ponovo pojavljuje sa suprotne strane.

Naredbe PLOT, CIRCLE i DRAW donekle su izmenjene, tako da prethodno proveravaju sadržaj specijalnih bejzik promenljivih XOS, YOS, XRG i YRG, kojima možete definisati koordinatni početak i razmeru koordinatnog sistema. Tako na primer, ako uradite LET xos=128: LET yos=88, onda će nadalje koordinatni početak uvek biti u centru ekrana. Recimo, naredba PLOT 0,0 će nacrtati tačku u centru ekrana.

Postoji i naredba FILL za bojenje proizvoljnih zatvorenih figura, naredba ALTER za trenutnu promenu atributa pojedinih karaktera na ekranu i naredba USING, pomoću koje se mogu štampati numerički podaci u zadatom formatu.

Procedure i programski ciklusi

Na kraju, postoji i grupa naredbi koje doprinose boljoj struktuiranosti programa. Naredba ON, u kombinaciji sa GO TO ili GO SUB, omogućuje tzv. „promenljiv skok“, na način kako se to izvodi u nekim drugim programskim jezicima, recimo u FORTRAN-u. Na primer, naredba GO TO ON h; 95, 75, 200 znači da će skok izvršiti na liniju 95 ako je h=1, na liniju 75 ako je h=2, a na liniju 200 ako je h=3.

U beta bejziku je moguće pozivanje potprograma i bez upotrebe linijskog broja, već navođenjem imena potprograma. To su takozvane procedure (otprilike nešto slično kao u PASCAL-u). Princip je isti kao kod DEF FN funkcija. Prosto definišete neku proceduru (naredbom DEF PROC), a onda je kasnije pozivate naredbom PROC.

I konačno, na raspolaganju vam stoje programski ciklusi sa kontrolom izlaza (tzv. DO-LOOP petlje). U standardnom bejziku postoje samo FOR-NEXT petlje, koje se izvršavaju određen broj puta. Međutim, ponekad ne znamo unapred koliko puta neki postupak treba ponoviti (npr. prilikom iteracije ponavljanje se vrši sve dok rezultat ne počne da se ponavlja. Istina, ovakvi problemi se mogu rešiti upotrebom GO TO naredbe, ali takvo rešenje nije uvek elegantno. Naredbe DO i LOOP pružaju daleko više mogućnosti u tom pogledu. Kada računar naiđe na naredbu DO, on jednostavno zapamti njenu adresu, a onda nastavi izvršenje programa koji sledi. Kada naiđe na naredbu LOOP, on se vrati na DO i radi sve iznova. Kontrolu izlaska iz petlje obavljamo pomoću naredbi WHILE i UNTIL (kao i obično, nije loše znati engleski). Recimo, možete staviti DO WHILE x>0, ili, što je potpuno isto, DO UNTIL x<0, i petlja će se izvršavati samo ako je x pozitivno. Kontrolu možete vršiti i na kraju petlje (LOOP WHILE ili LOOP UNTIL), a takođe i bilo gde unutar petlje (EXIT IF).

Osim toga, znatno su povećane mogućnosti i FOR-NEXT petlji, uvođenjem naredbe ELSE, koja je, inače, prisutna u gotovo svim drugim verzijama bejzika.

Nove funkcije

Što se tiče novih funkcija, samo ćemo pomenuti neke najinteresantnije. Funkcije INSTRING vrši pretraživanje datog alfanumeričkog niza nalazeći zadati podniz karaktera. U kombinaciji sa funkcijom MEMORY, koja prosto vraća celu memoriju računara u obliku jednog niza, moguće je jednostavno pretraživanje memorije, u cilju nalaženja raznih adresa (npr. adresa pojedinih varijabli i slično).

Funkcija MOD obavlja deljenje dva broja i vraća ostatak tog deljenja (suprotno od funkcije INT). Postoje i binarne logičke funkcije AND, OR i XOR, kao i funkcije za pretvaranje decimalnih brojeva u heksadecimalne ili binarne i obrnuto, itd.

Naravno, nema potrebe da detaljno objašnjavamo rad svih naredbi i funkcija, jer nam nije cilj da prepisujemo upustvo koje ćete dobiti uz program, već da vam pružimo što potpuniju informaciju o onome što ćete dobiti ako se ipak odlučite da svojih 9K (deviznih!) poverite beta bejziku. A naš savet je da to svakako učinite.

Jovan Skuljan

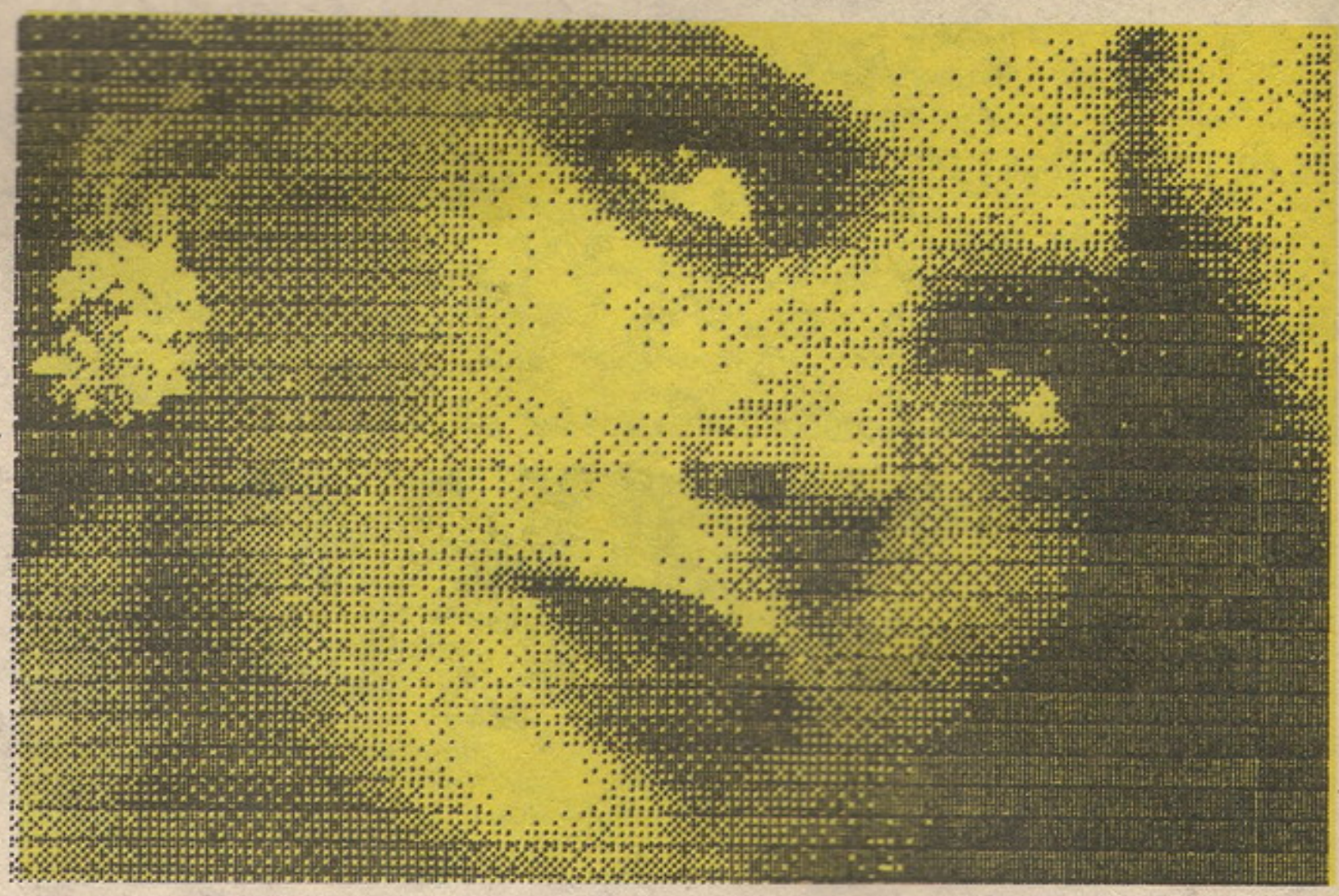
SIMON'S BASIC

Da li ste i sami, gledajući neku majstorski urađenu videoigru, poželeli da programirate tako efektno? Standardni bejzik je dosta siromašna alatka, a od brzih i moćnih mašinskih programa deli vas poznavanje procesora, metoda adresiranja, raspodele memorijskog prostora . . . Bez učitelja i jasnih priručnika na maternjem jeziku teško se stiže do savršenstva mašinskog programiranja. Zato preporučujemo da po naučenim osnovama programiranja na bejziku pokušate da radite u proširenom (SIMON'S) bejziku koji omogućava da bez zalaženja u detalje hardverske realizacije vašeg računara koristite većinu njegovih mogućnosti. Ostajete na nivou višeg programskog jezika, raspolažete moćnim interpretatorom, oslanjate se na već poznati bejzik, a pruža vam se i mogućnost struktuiranog programiranja, kontrole učitavanja i izdavanja podataka, svetlosnih efekata i skrolovanje ekrana, jednostavnog crtanja likova i animacije uz korišćenje zvuka. Da ne nabrajamo dalje — primamljive mogućnosti i za profesionalne programere.

Pomoć u pisanju programa

Većina računara poseduje funkcijske tastere koji pojednostavljuju upotrebu često korišćenih naredbi. U proširenom bejziku svakom funkcijskom tasteru može da se dodeli proizvoljna instrukcija ili niz znakova naredbom KEY. Na primer, KEY 1, „LIST—100" dodeljuje tasteru F1 naredbu LIST—100. Pritiskom na RETURN ona se izvršava. Posle KEY 3, „RUN"+CHR$(13) automatski se startuje program pritiskom na F3. (CHR$(13)=RETURN). Naredba DISPLAY prikazuje na ekranu sadržaje svih funkcijskih tastera. Programer ima na raspolaganju i automatsku numeraciju (AUTO) i prenumeraciju (RENUMBER) programskih redova sa datim korakom. Na žalost, adrese skokova kod GOTO i GOSUB naredbi (desne adrese) ostaju iste, pa naredba RENUMBER ima još dve neprijatne specifičnosti koje mogu da zbune programera.

Programeri u bejziku često razvijaju samostalne rutine koje nalaze primenu u mnogim programima. Naredba MERGE omogućava da programe koje koristite često čuvate na disketi ili kaseti i, po potrebi, dodajete ih programima koji se nalaze u memoriji. Pri tome morate voditi računa o tome da linijski brojevi rutina koje dodajete budu veći od brojeva programskih redova u memoriji i da se razlikuje definisanje promenljivih. Iste promenljive u različitim delovima programa mogu kasnije da izazovu teške asinhrone greške. (Greške mogu biti sinhrone i asinhrone sa programom koji ih je izazvao. Sinhrona greška nastaje dok je program u toku i može biti otkrivena pre no što se izvrši sledeća instrukcija. Deljenje nulom je primer takve greške. Asinhrona greška nastaje po isteku izvesnog vremena od događaja koji je prouzrokovao. Većina U/I grešaka su ovog tipa.) Najbolji put da se izbegnu problemi u vezi sa naredbom MERGE je jedinstveno struktuiranje svih programa.



Konstruktori na popravnom ispitu: Prošireni bejzik „Simon's Basic" znatno olakšava rad sa grafikom i zvukom na „komodoru 64"

Kod izdavanja bejzik listinga CMB—64 je veoma nekomforan. Program se može izlistati komandom LIST, brzina listanja se kontroliše tasterom CTRL, a zaustavljanje se vrši sa RUN/STOP. Posle zaustavljanja, listanje se može nastaviti jedino ponovnim LIST. Prošireni bejzik nudi nove komande za upravljanje listanjem. (KOMANDA — COMMAND može se upotrebljavati jedino u neposrednom režimu rada, za raliku od NAREDBE — STATEMENT koja je sastavni deo programa. Kod jezika koji zahtevaju prevođenje, a ne interpretaciju, ova razlika je očiglednija.) Može se listati program u blokovima, zadavati brzina listanja, zahtevati isticanje svih naredbi proširenog bejzika prilikom listanja i izdavanje svih programskih redova koji sadrže neku ključnu reč ili azbučni podatak.

Sem toga, ovaj program pruža pomoć u traženju grešaka i mogućnost obrade grešaka u programu. Sa TRACE i RETRACE prate se brojevi programskih redova naredbi koje se izvršavaju, sa DUMP dobija prikaz trenutnog sadržaja svih korišćenih promenljivih, dok se sa COLD vrši reset glavnog dela memorije, pri čemu sam SIMON'S BASIC ostaje netaknut. Regeneracija programa izbrisanog sa NEW ili COLD postiže se komandom OLD. Korišćenjem naredbe ON ERROR: GOTO može se predati upravljanje određenom delu programa u slučaju pojave greške, a sa NO ERROR ukinuti dejstvo prethodne naredbe. Ako želite da zaštitite neki deo svog programa, upotrebite naredbu DISAPA. Svi programski redovi koji počinju sa DISAPA po upotrebi komande SECURE Ø postaju nevidljivi. Pretpostavimo, na primer, da red 20 sledećeg programa treba da se zaštiti:

```
10 INPUT „UNESITE SHIFRU"; A$
20 IF A$=„NABUKODONOSOR" THEN 40
30 PRINT „SHIFRA NIJE KOREKTNA":
   GOTO 10
40 PRINT „SHIFRA JE KOREKTNA —
   UNESITE SVOJE IME"
```

To se postiže naredbom DISCAPA ispred teksta reda 20. Ako, zatim, sa LIST prikažete ovaj programski red, utvrdićete da su iza DISCAPA stavljene četiri dvotačke.

```
20 DISAPA :::: IF A$=„NABUKODO-
   NOSOR" THEN 40
```

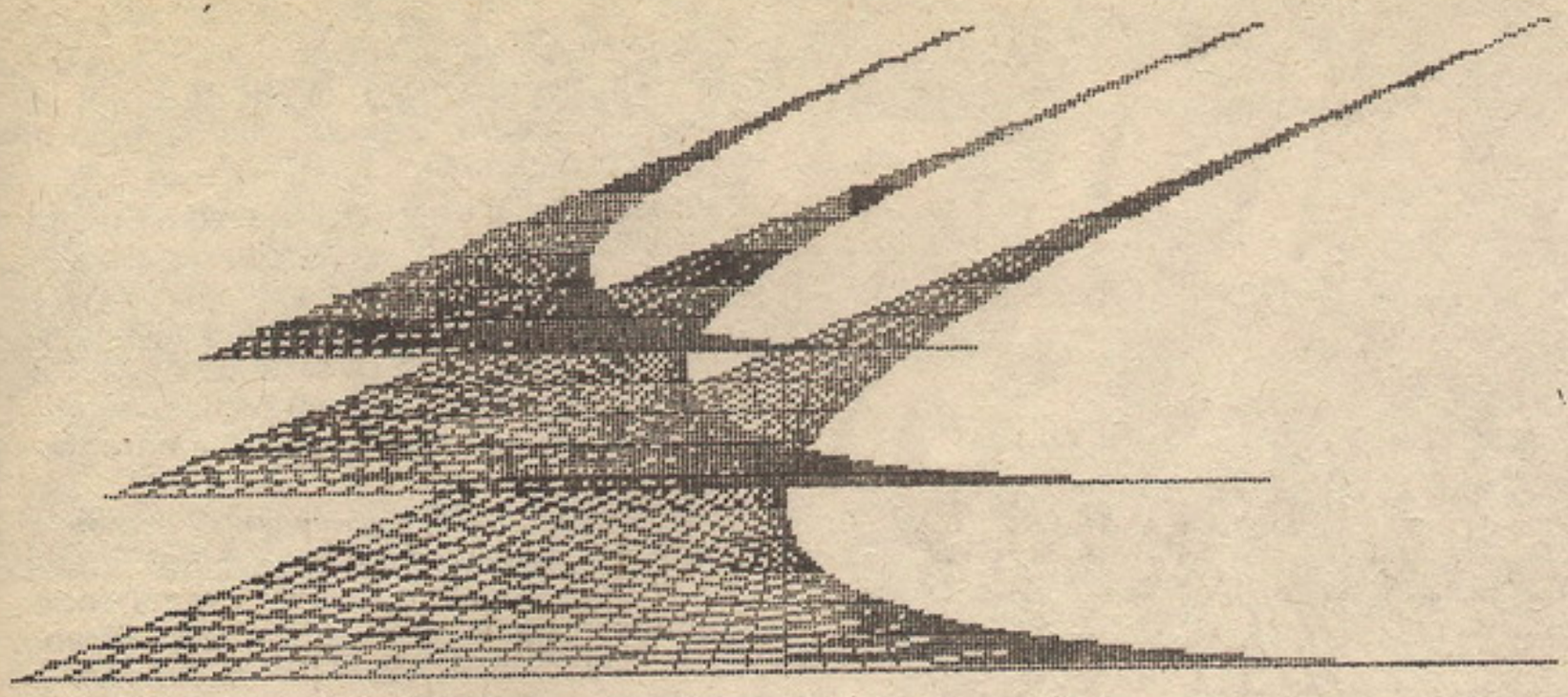
Posle upotrebe komande SECURE Ø videće se listanjem samo broj zaštićenog programskog reda, ali ne i njegov sadržaj. Nedostatak ovog metoda je što se sadržaji zaštićenih redova ne mogu vratiti. Stoga se preporučuje da se zaštićeni programi čuvaju u dve verzije: sa DISAPA naredbom i posle SECURE 0.

Kako je moguće da jedan red postane nevidljiv a ipak radi? Trik je u tome da se na početku odgovarajućeg reda stavlja bajt sa sadržajem 0, na osnovu kojeg ovaj programski red ne biva prikazan, a zatim četiri dvotačke, na osnovu kojih se ipak obrađuje.

Struktuirano programiranje

Mogućnost struktuiranog programiranja predstavlja jednu od najjačih strana proširenog bejzika. Umesto klasične FOR . . . NEXT petlje, koja omogućava jedino brojačke cikluse, mogu da se koriste REPEAT . . . UNTIL petlje koje omogućavaju izlazak iz ciklusa po zadovoljenju izlaznog kriterijuma bilo da je to broj izvršavanja ciklusa, bilo postignuta tačnost ili neki treći kriterijum. Sem toga, korišćenjem instrukcija LOOP . . .EXITIF . . .END LOOP kriterijum izlaska iz ciklusa može se postaviti u sredini petlje. Naredbama IF . . .THEN . . .ELSE I RCOMP . . .ELSE postiže se pregledniji zapis logičkih uslova.

Delovi programa izdvajaju se u procedure naredbama PROC i ENDPROC, a pozivaju sa CALL i EXEC, koje odgovaraju problematičnim GOTO i GOSUB. Kako se ovde prenos upravljanja ne predaje programskom redu sa određenim brojem, već delu programa sa određenim imenom (simboličkom adresom), to više nema ni opisa problema sa RENUMBER. Uz ovo postoji i



```
0 colour6,6
10 hires 1,6
20 for x=0 to 320 step 8
25 :  x1=x*3/4:x2=x/2
30 :  m1=140/320*x
40 :  m2=100/240*x1
50 :  m3= 70/160*x2
60 :  line 320-x ,m1,x,140,1
70 :  line 260-x1,m2,20+x1,100,1
80 :  line 200-x2,m3,40+x2, 70,1
90 next x
```

mogućnost definisanja lokalnih (LOCAL) i globalnih (GLOBAL) promenljivih svake procedure, tako da SIMON'S BASIC, poput kobola i paskala, omogućava kreiranje dobro struktuiranih i uzornih programa.

Obrada podataka

Pored aritmetike pokretnog zareza, prošireni bejzik omogućuje i celobrojnu deobu (DIV), dobijanje ostataka pri celobrojnoj deobi (MOD), izdavanje razlomljenog dela broja dobijenog izračunavanjem aritmetičkog izraza, ekskluzivne disjunkcije (EXOR) i rad sa binarnim i heksadekadnim brojevima.

Binarni zapis i EXOR omogućavaju nam jednostavnu izmenu nekog bita na određenoj memorijskoj lokaciji. Recimo da bit 7 memorijskog registra 53250 treba da bude promenjen. Umesto da proveravamo da li je tu bila 0 ili 1 i da, zatim, pomoću OR uključujemo (dekadno 128) bit na sedmoj poziciji bajta ili još komplikovanije, isključujemo pomoću AND i komplementa, upotrebićemo samo

POKE 53250, EXOR
(PEEK(53250,%01000000).

Iskusniji programeri u bejziku koriste sistemske rutine čije se adrese zapisuju uglavnom heksadecimalno. Ako ste ranije želeli da koristite npr. „hladni start" trebalo je prvo dekadno obračunati sistemsku adresu $FCE2 i zatim izvršiti SYS 64738. Sada to možete jednostavnije sa SYS $FCE2.

Sa alfanumeričkim podacima (stringovima) radi se takođe znatno komfornije. Možete umetati jedan string u drugi (INSERT), ispisivati jedan preko drugog (INST), izdavati poziciju stringa u nekoj azbučnoj promenljivoj (PLACE) i umnožavati string (DUP).

PLACE je vrlo efikasna naredba koja se sa standardnim bejzikom teško može simulirati. Uz kombinaciju sa INST-om mogu se jednostavno postići vrlo korisni efekti. Primer: za razliku od američke notacije, gde se razlomljeni deo broja razdvaja tačkom, kod nas se koristi zarez. Uz pomoć PLACE možemo brojeve izdavati na naš način.

```
10 INPUT „IZNOS:";B
20 B$=STR$(B)
30 P=PLACE     (".",BS)
40 B$=INST (",",B$,P—1)
50 PRINT „IZNOS:";B$
```

Komunikacija sa periferijom

Mogućnost programske kontrole ulaznih podataka zasniva se na naredbama FETCH, INKEY i RESET. Prva obezbeđuje prihvatanje jedino podataka koji su prethodno specificirani između znakova navoda. Na primer, vrlo se često dešava da u jednom programu postavljamo D/N pitanja. Sa naredbom FETCH problem se rešava mnogo elegantnije nego sa uobičajenim GET:

```
10 PRINT „PROGRAM ZAVRŠEN (D/N)?"
20 FETCH „DN", 1,X$
30 IF X$=„D" THEN END.
```

INKEY ispituje pritisnutost funkcijskog tastera. Rezultat je broj funkcijskog tastera koji se memoriše u jednoj unapred određenoj promenljivoj. RESET omogućava vraćanje pokazivača podataka na početak bilo koje DATA naredbe.

Uobičajena komunikacija sa diskom je do maksimuma pojednostavljena, a nude se i naredbe za memorisanje sadržaja ekrana niske rezolucije na disketu ili kasetu (SCRV). Ovo je naročito korisno kod rada sa tzv. maskama ekrana koje se memorišu i prema potrebi prikazuju na ekranu. Učitavanje se vrši naredbom SCRLD, a izdavanje na štampač pomoću HRDCPY. Na žalost, ne postoji naredba za memorisanje ekrana visoke rezolucije ali se zato on naredbom COPY može izdati na štampač.

Izdavanje podataka

Kada smo na početku ovog teksta govorili o pisanju lepih programa, nismo mislili na njihovu kratkoću i eleganciju već na njihov profesionalni spoljašnji izgled. SIMON'S BASIC pruža naredbe za formatizovanje, signalizaciju, punjenje i skrolovanje ekrana. Pod formatizovanim izlazom podrazumeva se sređeno pozicioniranje komentara, računa i slično, koje biste želeli da učinite vidljivim na ekranu prema određenim kriterijumima. Sem standardnih mogućnosti naredbe PRINT, tu su još centriranje teksta, izdavanje podataka poravnatih u odnosu na decimalnu tačku, pozicioniranje izdavanja podataka, izdavanje broja reda u kojem se nalazi cursor (pozicije od koje bi trebalo da se izvrši sledeće štampanje) i pravljenje pauze u izvršavanju naredbi sa ili bez komentara.

Može se postići treptanje na ekranu svih karaktera ispisanih određenom bojom i treptanje rama ekrana (BORDER-a). Određivanje boje pozadine i rama ekrana postiže se jednostavno sa COLOUR bb, bp gde su: bb — boja border-a i bp — boja paper-a (pozadine slova).

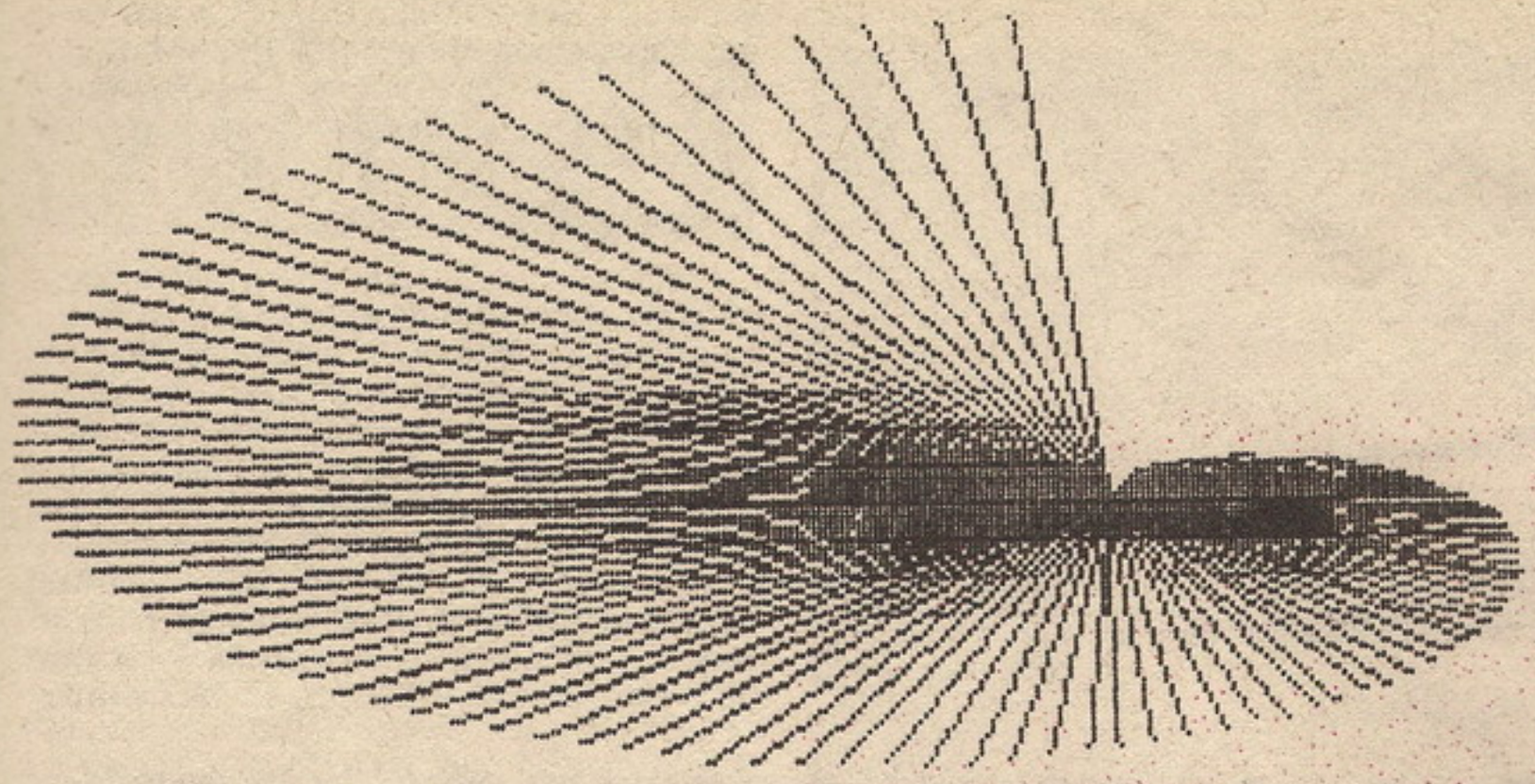
Jednom instrukcijom može se popuniti polje na ekranu znakom čiji je kôd zadat, određenom bojom ili kopiranjem sadržaja nekog drugog polja. Sem toga, na raspolaganju vam je mogućnost skrolovanja (pomeranja) ulevo, desno, gore ili dole određenog dela ekrana.

Grafika i muzika

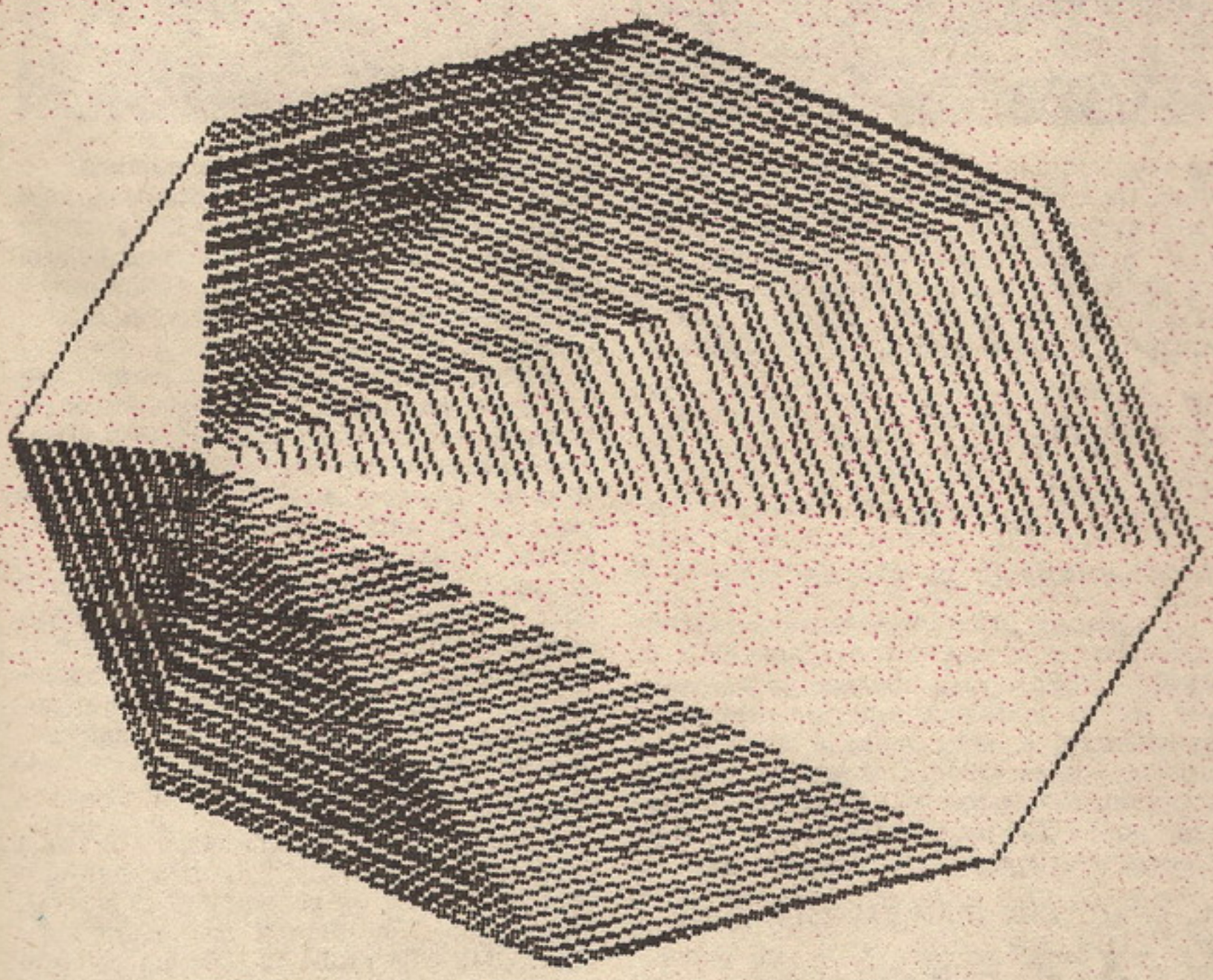
Sprajt je lik koji programer kreira sam i koji nezavisno od sadržaja ekrana ili drugih sprajtova, može da se pomera po ekranu. Zahvaljujući ugrađenom videocontroleru 6567, CMB—64 može da upravlja sa osam takvih likova. Svaki od njih može se razlikovati po veličini, boji i prioritetu od drugih sprajtova i pozadine. Rad sa sprajtovima u proširenom bejziku je veoma jednostavan. Oblik sprajta definišemo pomoću obrasca koja se sastoji od 21 reda sa po 12 (kod MC) ili 24 (kod HGR) pozicija.Osim oblika, boje i tipa (MC, HGR), svakom sprajtu dodeljujemo i prioritet na osnovu kojeg je on iza ili ispred pozadine. Za tako definisane sprajtove, dalje navodimo veličinu i pravac pomeranja. Sprajtove je moguće predstaviti u normalnoj veličini, dvostruko šire, dvostruko više ili dvostruko uvećane i

REČNIK SIMON'S BASICA

$	%	ANGL	ARC	AT	AUTO	BFLASH
BEFLASH0	BLOCK	CALL	CENTRE	CGOTO	CHAR	CHECK
CIRCLE	CMOB	COLD	COPY	CSET0/1	CSET2	DELAY
DESIGN0/1	DESIGN2	DETECT	DIR	DISAPA	DISK	DISPLAY
DIV	DOWN	DRAW	DUP	ELSE	ELSE	ENDLOOP
ENDPROC	ENVELOPE	ERRLN	ERRN	EXEC	EXIT	EXOR
FCHR	FCOL	FETCH	FILL	FIND	FLASH	FRAC
GLOBAL	HICOL	HIRES	HRDCOPY	IF	INSERT	INST
INV	JOY	KEY	LEFT	LIN	LINE	LOCAL
LOOP	LOWCOL	MEM	MERGE	MMOB	MOBOFF	MOBSET
MOD	MOVE	MULTI	MUSIC	NOERROR	OFF	OLD
ONERROR	OPTION	OUT	PAGE	PAINT	PENX	PENY
PLACE	PLAY	PLOT	POT	PROC	RCOMP	REC
REPEAT	RENUMBER	RESET	RETRACE	RIGHT	RLCOMB	ROT
SCRLD	SCRSV	SECUREO	TEST	TEXT	THEN	TRACE
UNTIL	UP	USE	VOL	WAVE		



```
0 colour 10,10
5 hires 6 ,10
10 for x=5 to 358 step 3
20 : angl 220,125,x,x/1.4,x/3.8,1
30 next
```



```
0 hires 1,5
10 for x=5 to 100 step 2
20 : arc 100+x*.8,100,315, 90,45,x*1.2,x,1
30 : arc 100+x*.8,100,135,270,45,x*1.2,x,1
40 next x
50 arc 180,100,0,360,45,120,100,1
```

po širini i po visini. Programski možemo pripremiti ispitivanje i ispitivati da li je došlo do sudara između dva sprajta ili sprajta i pozadine.

Veliki grafički potencijali vašeg računara inače teško dostupni iz standardnog bejzika sada mogu, bez zalaženja u finese realizacije, obilato da se koriste zahvaljujući gotovim naredbama za crtanje. Svi prilozi uz ovaj tekst urađeni su korišćenjem naredbi proširenog bejzika u vrlo kratkim programima.

Naredbom HIRES prelazi se iz modaliteta za tekst u finu grafiku i isključuje memorija grafike — posle izdavanja ove naredbe programer pred sobom vidi prazan ekran. Ovo je važno da se zna jer su povraci u modalitet u tekst u toku jednog programa veoma česti. Naravno, naredba HIRES se može primeniti i dok se prikazuje grafika. Prilikom isključivanja slike vraćaju se svi bajtovi memorije grafike i vidi se samo boja pozadine iz video rama. Još jedna napomena. Mada je moguće ovu naredbu primeniti i van programa, ima je smisla upotrebiti jedino kao sastavni deo programa. MULTI na sličan način uključuje multikolor grafiku, a HICOL crta bez uključivanja boje. LOW COL menja boju prilikom crtanja, dok NRM omogućava povratak u modalitet za tekst.

Naredba CEST izvodi niz različitih i, na prvi pogled, nevezanih funkcija. Pomenimo najinteresantniju — kreiranje sopstvenog seta znakova. Dakle, ako želite da programirate koristeći ćirilicu, pruža vam se i ta mogućnost. Ako posedujete matrični štampač, možete dobiti i na papiru odgovarajuće poruke ispisane vašom garniturom znakova.

Naredbe crtanja u proširenom bejziku omogućavaju da prikažete, obrišete ili invertujete pravu liniju, pravougaonik, elipsu, luk i radijus elipse i da zatvorene površi ispunite odabranom bojom. Sem toga, postoji mogućnost da nacrtate proizvoljnu sliku na ekranu, koju, zatim, možete da rotirate i uvećavate. Pogledajte grafike koje prate ovaj tekst. Sve su one dobijene korišćenjem naredbi za crtanje.

Za korišćenje zvuka SIMON'S BASIC pruža sledeće instrukcije: VOL podešava glasnoću za sva tri glasa, WAVE određuje oblik talasa za određeni glas (kanal), ENVELOPE definiše zvučnu sliku tonova koje treba odsvirati, dok MUSIC daje redosled tonova muzičkog komada koji se svira sa PLAY. Na žalost, nismo u mogućnosti da vam, kao kod opisa naredbi grafike, prikažemo na primerima kako izgleda njihovo dejstvo.

SIMON'S BASIC je očigledno, veoma moćan i dobro urađen viši programski jezik. Iz priloga možete videti da koristi više od stotinu novih instrukcija, pa jednim tekstom nije moguće objasniti detalje njegovog korišćenja. Programeru koji se odluči za ovo proširenje stoji na raspolaganju udžbenik sa mnoštvom primera koji i početnike uspešno uvodi u tajne programiranja. Tvorac ovog jezika Dejvid Sajmon (David Simon) još uvek je tinejdžer. Znamo da mnogi njegovi vršnjaci koji čitaju ove redove sve svoje slobodno vreme posvećuju računarima i verujemo da su u stanju da naprave nešto slično. Pozivamo ih da pokušaju. Sistemski softver nije ništa manje zanimljiv od igara. Naprotiv!

Nevenka Spalević

halo, da li je to mašinac?

Majstorije na računaru

Spectrum

Kraće mašinske rutine u okviru bejzik programa mogu da budu od izuzetne koristi. Natpisi sa uvećanim slovima u programima sa kasete koja se dobija uz „spektrum“ deluju, ruku na srce, izuzetno efektno. Poslove ovakve prirode obavljaju mašinski programi. Naravno, ne zato što bejzik nije u stanju da to uradi. „Spektrumov“ bejzik bi, bar u principu, mogao da obavi sve što i mašinski program, samo to ne bi imalo smisla, jer bi izvršenje programa trajalo beskrajno dugo, a i sam program bi zauzeo suviše dragocene memorije. Operativni sistem računara „spektrum“ dozvoljava slobodno šetanje iz bejzika u mašinski program i nazad. Problem, međutim, nastaje kada pri tom treba proslediti i neke parametre.

Pozivanje mašinskih programa vrši se funkcijom USR. Međutim, većina mašinskih rutina zahteva i neke ulazne podatke. Na primer, program za štampanje teksta treba da „zna“ koji će tekst biti štampan, na kom mestu ekrana, kolika će biti širina i visina slova i razmak između njih. Za ovo se koristi čitav jedan bejzik potprogram koji sve te podatke prosleđuje mašinskoj rutini pomoću naredbe POKE. Ovakav postupak, iskreno govoreći, ne odiše preteranom elegancijom. Zar ne bi sam mašinski program mogao da obavi to prikupljanje podataka? Mogao bi, razume se, kada bi znao gde da ih traži. Samo, nemojte pomisliti da je rešenje u tome što ćemo mašinskoj rutini poslati adrese, umesto samih ulaznih podataka. Jer to bi se, naravno, opet svelo na nekakav POKE potprogram. Englez Majk Džems otkrio je kako se ovo postiže korišćenjem.

Spektrumove DEF FN funkcije

Kako se u memoriji „spektruma“ čuvaju programske linije sa naredbama za definisanje funkcija? Uzmimo, kao primer, jednostavnu funkciju za sabiranje dva broja:

10 DEF FN f (x, y)=x+y

Kada otkucate ovakvu liniju i pritisnete ENTER, ona će biti poslata na odgovarajuće mesto u memoriji tako što će svaka naredba, ili svaki znak iza linijskog broja, zauzeti po jedan bajt. Pored toga, iza svakog parametra funkcije, a to su ovde x i y između zagrada, biće umetnuto po šest bajtova. Prvi od tih šest bajtova uvek sadrži vrednost četrnaest, što je prosto oznaka koja govori računaru da upravo sledi neki broj u binarnom obliku. A pet bajtova koji slede iza bajta 14 služiće kasnije, u fazi izračunavanja funkcije f, za smeštanje stvarnih brojnih vrednosti parametara x i y.

Ako se sada, negde u programu, nađe funkcija, npr. FN f (2, 7) desiće se sledeće stvari:

— Proverite, proverite, mora da postoji neki bag!

- računar pretražuje bejzik program ne bi li pronašao liniju u kojoj je definisana funkcija f
- kada se takva linija pronađe, a u našem slučaju to će se desiti dosta brzo jer smo definisali funkciju u liniji 10, pronalazi se adresa prvog parametra iza otvorene zagrade; u našem slučaju to je parametar x
- adresa prvog parametra se smešta u sistemsku varijablu DEFADD (njena adresa je 23563)
- broj 2 se u binarnom obliku šalje u jedan plus pet bajtova iza parametra x, a broj 7 iza parametra y; na taj način je olakšano rutini za izračunavanje same funkcije da prikupi podatke i primeni na njih priložen zakon, tj. operaciju sabiranja u našem slučaju; na primer, kada dođe vreme da se pokupi vrednost broja x, ona će biti nađena počev od relativne adrese 2, mereno u odnosu na sadržaj DEFADD; vrednost y treba potražiti počev od relativne adrese 10

Da bismo videli kako to izgleda, promenićemo definiciju funkcije f, uvodeći jedan kratak mašinski program koji će simulirati operaciju sabiranja. Program ćemo smestiti u bafer za štampač, počev od adrese 23296. Dakle, linija 10 neka sada bude:

10 DEF FN f (x, y)=USR 23296

Sta će se desiti kada upotrebite funkciju

FN f(2, 7)? Naravno, sve ono što smo već rekli, s tim što će umesto poziva pomenute rutine za sabiranje, doći do skoka na naš mašinski program na adresi 23296.

Iz načina na koji smo definisali funkciju f, vidi se da će njen rezultat uvek biti neki broj između nule i 65535, jer funkcija USR, zapravo, ima takav rezultat — broj od 16 bita. Ali, koji je to broj u stvari? Svakako, ne neki „bez veze". To je onaj broj koji se trenutno našao u registarskom paru BC u trenutku povratka iz mašinskog programa u bejzik. Razume se, često vas uopšte ne interesuje šta se zateklo u registru BC po završetku mašinske rutine. Tada takve programe i ne izvršavate sa PRINT USR . . ., već sa RANDOMIZE USR . . . Ali, ako mašinski program već služi za nalaženje nekog broja koji ne zauzima više od 16 bita, treba samo pre povratka u bejzik smestiti taj broj u BC i tako dobiti rezultat direktno.

Prenošenje parametara

Zato ćemo program za sabiranje napisati tako da radi korektno samo sa osmobitnim brojevima (0—255), jer će rezultat takvog sabiranja sigurno stati u 16 bita.

Program počinjemo tako što adresu parametara iz DEFADD prenosimo u registar IX. Zatim uzimamo broj x, a to je bajt IX+4, i smeštamo ga u registar L. Naredba LD L, (IX+4) upravo i znači: napuni registar L sadržajem sa adrese IX+4. Slično, u registar E uzimamo broj y. Ako vam još uvek nije jasno zašto brojeve x i y nalazimo baš na tim adresama, upišite sami brojeve, recimo 2 i 7, u prazne bajtove iza x i y, prema uputstvu koje smo dali. Videćete da se broj 2 zaista nalazi na relativnoj adresi 4, a broj 7 na adresi 12.

Program se dalje odvija tako što se u registre H i D ubacuju nule, čime je postignuto da sada čitavi registarski parovi HL i DE sadrže brojeve x, odnosno y. Ovaj manevar izveli smo zato što u mašinskom jeziku postoji naredba za sabiranje registarskih parova HL i DE, pa ćemo je mi upotrebiti direktno. Razume se, to nije jedini način da obavimo posao koji želimo, ali za sada je bitno samo to da program radi. Posle naredbe sabiranja, rezultat iz HL prenosimo u BC, registar po registar, i onda, konačno, naredbom RET vršimo povratak u bejzik.

Dajemo čitav mašinski program, zajedno sa adresama i kodovima svih naredbi. Obavezno pokušajte da ubacite taj program u računar, pogotovo ako mislite da ne umete da pravite mašinske programe. Videćete da to nije nikakav problem. Kada ubacite sve kodove na njihova mesta i otkucate bejzik liniju 10, možete da eksperimentišete sa funkcijom f, koristeći je kao i bilo koju drugu funkciju. Na primer PRINT FN f (2, 7).

Prenošenje nizova

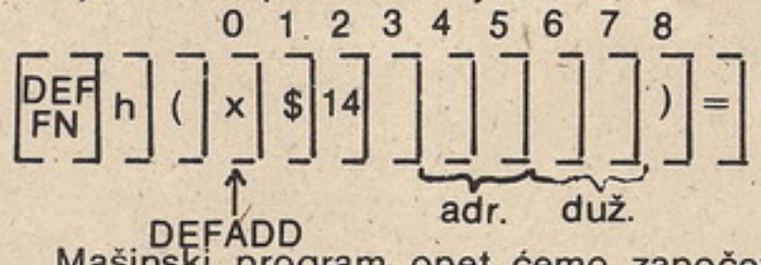
Sada već imate predstavu o tome kako se ulazne veličine mogu jednostavno prenositi u mašinsku rutinu — treba samo definisati jednu funkciju koja poziva mašinski program, čiji parametri su upravo same ulazne veličine. Mašinski program treba modifikovati tako da sam pronalazi potrebne veličine, koristeći sistemsku varijablu DEFADD. Još vas jedino može zanimati problem prenošenja slovnih nizova.

```
23296  LD   IX, (23563)  221,42,11,92
23300  LD   L, (IX+4)    221,110,4
23303  LD   E, (IX+12)   221,94,12
23306  LD   H,0          38,0
23308  LD   D,0          22,0
23310  ADD  HL,DE        25
23311  LD   B,H          68
23312  LD   C,L          77
23313  RET               201
```

Opet ćemo uraditi jedan prost primer: program koji prebrojava koliko se puta u nekom nizu pojavljuje, recimo, slovo „H". Definisaćemo zato funkciju:

10 DEF FN h(x$)=USR 23296,

pri čemu smo opet odlučili da smestimo mašinski program u bafer za štampač. Ali, pre nego što pređemo na sastavljanje samog programa, treba videti šta će se desiti sa konkretnom vrednošću niza kada bude došao red na izračunavanje funkcije h. I u ovom slučaju, kao i ranije, programska linija 10 se smešta u memoriju tako što se iza parametara, a u našoj funkciji figuriše samo jedan parametar, x$, umeće šest bajtova, od kojih je prvi opet 14. Takođe, u fazi izračunavanja funkcije, varijabla DEFADD sadrži adresu parametara — u našem slučaju to je x$. Sve je, dakle, isto kao da se radi o numeričkim parametrima, izuzev što onih pet bajtova iza x$, razume se, ne mogu služiti za smeštanje samog niza. Međutim, oni će sadržati sve potrebne informacije o samom nizu: prvi bajt je tu nevažan, drugi i treći će čuvati početnu adresu niza koji se, inače, nalazi negde u memoriji, a četvrti i peti bajt će sadržati dužinu niza, tj. ukupan broj slovnih znakova. Svaki karakter niza zauzima jedan bajt memorije. I adresa i dužina niza čuvaju se, opet, u obrnutom redosledu zapisivanja, kao što smo već govorili: prvo ide bajt manje težine, pa onda bajt veće težine.



Mašinski program opet ćemo započeti uzimanjem adrese iz DEFADD i njenim smeštanjem u registar IX. Onda ćemo pokupiti početnu adresu niza i smestiti je u registar HL, prvo uzimajući bajt manje težine u L, a onda bajt veće težine u H. Na potpuno isti način uzećemo i dužinu niza u DE. Nadalje će nam registar DE služiti kao brojač: čim ispitamo neko slovo niza, umanjićemo brojač za jedan. Kad, tako, stignemo do nule, biće to znak da je posao obavljen — ispitana su sva slova. A ukopan broj slova jednakih „H" smeštaćemo, prirodno, u registar BC, da bismo na kraju taj rezultat jednostavno vratili u bejzik. Očigledno, rezultat nikako neće biti veći od 16 bita, tj. 65535, pošto u vašem računaru i nema više mesta za smeštanje nekog niza.

Vrednost BC na početku stavljamo na nulu, kao znak da nismo našli još nijedno pojavljivanje slova „H", a onda ulazimo u petlju za ispitivanje svakog slova. Uzimamo slovo sa adrese HL i smeštamo ga u akumulator A. Naredba CP, skraćeno od „compare" znači: uporedi sadržaj akumulatora sa priloženim brojem — u našem slučaju to je 72, što odgovara kodu karaktera „H". Samo poređenje se inače svodi na oduzimanje broja 72 od akumulatora. Ako se dobije nula, to znači da je pronađen jedan karakter niza jednak „H" i treba uvećati brojač BC za jedinicu (naredba INC BC). Međutim, ako rezultat nije nula, treba preskočiti naredbu INC BC i odmah nastaviti od naredbe označene sa DALJE. Naredba JR NZ, DALJE upravo i znači: skoči na DALJE ako rezultat nije nula. Što se tiče dela programa koji počinje od simboličke adrese DALJE, tu se prvo proverava da li su ispitana sva slova: umanjuje se DE za jedinicu (naredba DEC DE), a onda se ispituje sadržaj registara D i E, primenom logičke ILI operacije. Naredbe LD A, D i OR E zajedno znače isto što i D ILI E. Ako je rezultat nula, to znači da su i D i E nule, pa je i DE nula. Dakle, ispitana su sva slova i može se obaviti povratak u bejzik (RET Z znači: povratak ako je nula). A ako nije nula, HL se uvećava za jedinicu da bi se ispitao sledeći znak i vrši se skok nazad u petlju.

Ubacite i ovaj program u računar da biste videli kako radi. Probajte, na primer, PRINT FN h („AHA-HA!"). Rezultat će, kao što i očekujete, biti jedinica. A PRINT FN h („AHA-HA!") daće dvojku.

Pokušajte da napravite bejzik program koji obavlja istu stvar i uporedite samo vremena izvršavanja!

```
90 INPUT a$
100 LET b=0: FOR n=1 TO LEN a$
110 IF a$ (n)="H" THEN LET b=b+1
120 NEXT n
130 PRINT b
```

Na primer, uradite DIM a$(500), pa onda izvršite bejzik program sa GO TO 100. Zatim primenite i vašu funkciju: PRINT FN h(a$). Primer je toliko ubedljiv da je svaki komentar suvišan.

Logičke operacije

Opisani metod prenošenja parametara primenićemo na još nekoliko jednostavnih programa, koji će vam svakako više koristiti od prethodnih primera. Uvešćemo neka poboljšanja u spektrumovu logiku.

Spektrum, kao što znate, raspolaže sa tri logičke operacije: AND, OR i NOT. Možda ih obilato koristite u svojim programima, a možda ih samo nezainteresovano posmatrate na tastaturi, ne znajući kako da ih upotrebite. Poći ćemo od pretpostavke da spadate u ovu drugu klasu — zainteresovanih posmatrača — i ukratko objasniti dejstvo logičkih operacija.

Matematička logika je veoma bliska svakodnevnoj logici u govornom jeziku. Uzmimo, na primer, dva iskaza.

1. Pada kiša
2. Pada sneg

Svaki od ovih iskaza može biti tačan ili ne, u zavisnosti od toga kakvo vreme vlada napolju. Ako je iskaz tačan, dodelićemo mu vrednost jedan, a ako je netačan, dodelićemo mu vrednost nula.

Sada ćemo uvesti operaciju **konjunkcije,** ili kratko „I" operaciju (sasvim slobodno možemo koristiti oznake iz našeg jezika). Formiraćemo sud:

„Pada kiša i pada sneg"

Očigledno, ovaj složen sud kaže da istovremeno padaju i kiša i sneg. On je, dakle, tačan samo ako su oba iskaza u njemu tačna. Ukoliko je bilo koji od tih iskaza lažan, i ceo sud će biti lažan. Drugim rečima, kad vam neko dođe i kaže: „Pada kiša i pada sneg", vi ćete pogledati napolje, padaju li zaista i kiša i sneg. Samo ako oba testa uspeju reći ćete: „U pravu si". Dovoljno je da ne uspe samo jedan, pa da kažete:

```
23296  LD    IX, (23563)  221,42,11,92
       LD    L, (IX+4)    221,110,4
       LD    H, (IX+5)    221,102,5
       LD    E, (IX+6)    221,94,6
       LD    D, (IX+7)    221,86,7
       LD    BC, 0        1,0,0
PETLJA LD    A, (HL)      126
       CP    72           254,72
       JR    NZ, DALJE    32,1
       INC   BC           3
DALJE  DEC   DE           27
       LD    A, D         122
       OR    E            179
       RET   Z            200
       INC   HL           35
       JR    PETLJA       24,243

LD     IX, (23563)  Adresa parametara iz DEFADD ide u IX
LD     A, (IX+4)    Bajt manje težine broja x ide u akumulator,
AND    (IX+12)      i dovodi se u konjunkciju sa odgovarajućim
LD     C,A          bajtom broja y.
LD     A, (IX+5)    Rezultat se smešta u registar C.
AND    (IX+13)      Potpuno isto se postupa i sa bajtovima veće
LD     B,A          težine.
RET                 Obavlja se konjunkcija, a rezultat šalje u B.
                    Povratak u bejzik. BC sadrži rezultat x I y.
```

```
 10 DATA 221,42,11,92,221,126,4
,221,166,12,79,221,126,5,221,166
,13,71,201
 20 DATA 221,42,11,92,221,126,4
,221,182,12,79,221,126,5,221,182
,13,71,201
 30 DATA 221,42,11,92,221,126,4
,47,79,221,126,5,47,71,201
 40 LET a=23296
 50 FOR n=0 TO 52
 60 READ b
 70 POKE a+n,b
 80 NEXT n
100 DEF FN i(x,y)=USR a: REM "I
"
110 DEF FN l(x,y)=USR (a+19): R
EM "ILI"
120 DEF FN n(x)=USR (a+38): REM
 "NE"
130 INPUT "x=? ";x;"y=? ";y
140 PRINT x;" I ";y;" = ";FN i(
x,y)

150 PRINT x;" ILI ";y;" = ";FN
l(x,y)
160 PRINT "NE ";x;" = ";FN n(x)
170 PRINT
180 GO TO 130
```

„Nisi u pravu". Matematički iskazano, operacija konjunkcije se može predstaviti sledećim relacijama:

1 I 1=1
1 I 0=0 KONJUNKCIJA JE TAČNA SAMO
0 I 1=0 AKO SU OBA ISKAZA TAČNA
0 I 0=0

Probajte sve ovo sa operacijom AND na Spektrumu, npr. PRINT 1 AND 1, itd.

Druga logička operacija je operacija **disjunkcije,** koju ćemo označavati sa „ILI". Primenjeno na naše „meteorološke" iskaze, to bi izgledalo ovako:

„Pada kiša ili pada sneg"

Sud će biti tačan bilo da pada kiša, bilo da pada sneg, a može biti i jedno i drugo. U logici, inače, postoji posebna operacija, tzv. „isključivo ILI", koja ne dopušta mogućnost da su oba suda istovremeno istinita. Nju bismo zapisali kao: „Ili pada kiša, ili pada sneg". Međutim, obična operacija disjunkcije lažna je jedino ako su lažna oba iskaza.

1 ILI 1=1DISJUNKCIJA JE TAČNA AKO JE
1 ILI 0=1BAREM JEDAN OD ISKAZA TAČAN
0 ILI 1=1
0 ILI 0=0

I ovo možete proveriti eksperimentišući sa operacijom OR na vašem računaru.

Konačno, postoji još i operacija **negacije,** koju ćemo označiti sa „NE". Na primer:

„Ne pada kiša"

Ovaj sud će biti tačan samo ako je lažan iskaz „Pada kiša" i obrnuto. Negacija, jednostavno, izvrće istinosnu vrednost nekog suda.

NE 1=0NEGACIJA JE TAČNA SAMO AKO JE
NE 0=1LAŽAN SUD KOJI SE NEGIRA

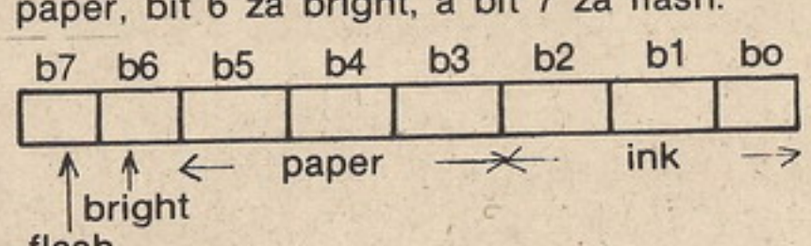
Operacija NOT na Spektrumu radi upravo tako.

Ipak, Spektrumove logičke operacije raspolažu sa prilično slobode u svemu tome — bez ikakve veze sa binarnim zapisom brojeva:

x AND y= x, ako y nije nula Ø, ako y jeste nula
x OR y= 1, ako y nije nula
x, ako y jeste nula
NOT x=1, ako je x nula Ø, ako x nije nula

Čak i ovako, operacije AND, OR i NOT su izuzetno moćne. Njihova upotreba će znatno pojednostaviti vaše programe, a, kao što vidite, uopšte nije teško koristiti ih.

Ipak, ne bi bilo loše raspolagati i „binarnim" logičkim operacijama, kao što su one u mašinskom jeziku. Ako baš i niste ubeđeni u to, podsetićemo vas, recimo, na funkciju ATTR, koja je prilično nespretno izvedena. Ona vam, jednostavno, vraća neki broj između nule i 255, a vi se dalje sami snalazite kako da odatle izvučete „ink", „paper", „bright" i „flash" informaciju o karakteru koji je ispisan na ekranu. Primenom binarne logike, stvar bi postala veoma prosta. Radi se o tome da je rezultat funkcije ATTR, u stvari, jedan binarni broj od osam bita. Bitovi 0, 1 i 2 nose ink informaciju, bitovi 3, 4 i 5 su odgovorni za paper, bit 6 za bright, a bit 7 za flash.



Ako želite da izdvojite ink informaciju, treba prosto, pogasiti (resetovati) sve bitove osim bo, b1 i b2. A bilo koji bit ćete sigurno dovesti na nulu ako primenite operaciju konjunkcije između tog bita i nule (bilo šta I nula=nula). S druge strane, konjunkcija sa **jedinicom** uopšte neće promeniti neki bit. Prema tome, ink se direktno dobija ako se atributni bajt dovede u logičku konjunkciju sa binarnim brojem 00000111. Slično, paper ćete izdvojiti u konjunkciji sa 00111000, ali rezultat još treba pomeriti udesno za tri mesta, da bi se dobio broj između nule i sedam.

Ovo „pomeranje" udesno, ili što je isto, pomeranje decimalnog zareza ulevo, ostvaruje se u bilo kom brojnom sistemu deljenjem sa osnovom sistema, a to znači dvojkom u našem slučaju. Svako deljenje sa dva pomeriće vaš broj udesno za jedno mesto, dok će množenje sa dva obaviti ekvivalentno pomeranje ulevo (isto se dešava u dekadnom sistemu kad delite i množite sa deset). Uzmite i sami nekoliko brojeva, napišite ih u binarnom obliku i onda pogledajte šta se dešava pri deljenju i množenju sa dva. Postaće vam jasno da paper možete dobiti trostrukim uzastopnim deljenjem prethodno dobijenog rezultata sa dva, a to je isto što i debljina sa osam ($8=2^3$). A što se tiče bright i flash informacije, njih ćete, razume se, dobiti dovodeći ATTR u konjunkciju najpre sa 01000000, odnosno 10000000, a onda delići rezultat sa 64 (2^6), odnosno sa 128 (2^7).

Binarna logika u bejziku

Da bismo uveli binarne logičke operacije, napisaćemo jednostavne mašinske rutine, koje će prosto koristiti postojeću „mašinsku" logiku, a za prenošenje parametara iskoristićemo već opisani metod pomoću DEF FN funkcija.

Uzmimo, najpre, operaciju „I". Definisaćemo funkciju oblika:

10 DEF FN i(x, y)=USR 23296

Ograničićemo se na slučaj kada brojevi x i y ne zauzimaju više od 16 bita, tj. nisu veći od 65535. U mašinskom jeziku ne postoji naredba AND koja bi radila sa 16-bitnim brojevima, ali to nije nikakav problem, pošto sasvim jednostavno možemo raditi posebno sa svakim bajtom. Na primer, vratimo se mikroprocesoru i zamislimo da se u registrima HL i DE nalaze dva 16-bitna broja, a mi želimo da u BC smestimo rezultat operacije HL „I" DE. To ćemo izvesti tako što u B pošaljemo vrednost H I D, a u C vrednost L I E.

Verujemo da će vam program koji dajemo biti sasvim jasan. Operacija AND (IX+n) direktno dovodi u konjunkciju sadržaj akumulatora i bajta na adresi IX+n, a rezultat se opet šalje u akumulator.

Dajemo kompletan bejzik program koji definiše sve tri logičke operacije i ilustruje njihovu primenu. Što se tiče mašinskih rutina, one su potpuno slične ovoj za funkciju „I". Jedino treba zameniti sve AND naredbe sa OR u slučaju „ILI" funkcije, ili sa CPL u slučaju funkcije negacije.

Možete, po želji, promeniti adresu od koje se mašinski program unosi u memoriju (linija 40), a možete i već formiran program iz bafera snimiti na kasetu, da biste ga kasnije unosili gde god to hoćete. Jedino ne zaboravite da promenite adrese u USR funkcijama.

O upotrebi operacije konjunkcije dosta smo govorili pominjući funkciju ATTR. Videli ste kako jednostavno možete resetovati pojedine bitove bilo kog broja. Slično tome, možete upotrebiti operaciju „ILI" da biste setovali neke bitove. Treba se setiti da je „bilo šta ILI 1=1", dok disjunkcija sa nulom uopšte neće promeniti dati bit. Na primer, ako želite da setujete samo bit 7, ostavljajući druge bitove kao što jesu, primenite „ILI" operaciju sa brojem 10000000 (128). Ako nekad budete u nedoumici, pogledajte podsetnik koji dajemo.

Jovan Skuljan

Šta Ejkorn nije rekao?

Majstorije na računaru

BBC/Electron

Neobjašnjene naredbe bejzika 2

Na samom početku treba da proverite da li je vaš računar uopšte opremljen bejzikom 2. Pritisnite BREAK i otkucajte REPORT, a zatim pritisnite RETURN. Ukoliko se na ekranu pojavi (C)1982 Acorn (a ne (C)1983 Acorn kao što je, štamparskom greškom, pisalo u „Računarima 2") — izvukli ste premiju. U protivnom, možete da kupite jedan EPROM 27128 (oko 110 DM u SR Nemačkoj) i da zamolite nekog drugog vlasnika BBC B koji ima pristup EPROM programatoru da za vas kopira svoj bejzik. Acorn će biti srećan da obavi ovu sitnu uslugu za „samo" tridesetak funti (vrlo nekorektan potez firme, uzgred rečeno). Vlasnike „electrona" ne moraju da more brige ovoga tipa: njihovi računari su obavezno opremljeni drugom verzijom bejzika, pa nema nikakve potrebe da menjaju ROM, ova promena bi, inače, bila povezana sa dosta problema, jer „electron" ima ROM od 32 Kb u jednom čipu, koji bi, dakle, morao da se prepisuje u relativno redak (i skup) EPROM 27256.

U bejziku 2 su, najpre, ispravljeni neki bagovi od kojih je patio bejzik 1. I inače neverovatno moćna naredba PLOT proširena je novim opcijama, dodata je jedna naredbi za rad sa datotekama i značajno unapređene karakteristike ugrađenog asemblera.

Kompletan asembler . . .

Jeste li ikada poželeli da napišete vaše ime u sredini mašinskog programa koji ste sastavili? Ukoliko sledite uputstvo za upotrebu, moraćete da zatvorite srednju zagradu i time „napustite" asembler, otkucate naredbu koja, otprilike, glasi: $P%=„Ime i prezime": P%=P%+14 i da zatim ponovo započnete asembler koristeći srednju zagradu i novu naredbu OPT. Uz bejzik 2 možete, ne napuštajući asembler, da otkucate EQUS „Ime i prezime" i nastavite sa pisanjem programa.

Naredbe tipa EQU se ne obraćaju direktno mikroprocesoru, radi se o informativnim naredbama koje se, ipak, razlikuju od OPT ili ORG. EQUB 10, na primer, ubacuje bajt 10 u mašinski program na mestu koje u tom trenutku pokazuje promenljiva P%. Na taj način možete sasvim jednostavno da definišete neki bajt koji ne mora da ima smisao neke mašinske instrukcije, to može da bude deo neke tabele, kôd zvanično nepostojeće instrukcije (dosta ovih instrukcija dobijate ako vaš 6502 zamenite „modernijim" 65C02) ili, jednostavno, neki marker.

EQUB, dakle, definiše jedan bajt mašinskog programa. EQUS, naredba koju smo maločas upoznali, omogućava definisanje čitavog stringa, odnosno niza slova proizvoljne dužine. Treba napraviti jasnu razliku između naredbe EQUS i komentara u okviru mašinskog programa: dok komentar biva jednostavno ignorisan od strane asemblera tako da ne ostavi nikakav trag u rezultujućem kodu, EQUS generiše niz bajtova i tako produžava mašinski program.

Osim EQUB i EQUS, konstruktori bejzika 2 su predvideli i naredbe EQUW i EQUD. EQUW omogućava smeštanje binarne reči (dva bajta) u memoriju na način koji odgovara konvenciji osmobitnih mikroprocesora. EQUW &1234, na primer, umeće bajtove &34 i &12 u mašinski program, omogućavajući, tako, jednostavnu promenu adresa JMP i JSR naredbi i slične efekte. EQUD je naredba koja će biti korišćena daleko ređe: ona u memoriju smešta dve reči, odnosno četiri bajta. EQUD &12345678, na primer, smešta u memoriju bajtove &78, &56, &34 i &12. Ovaj format odgovara smeštanju BBC-jevih celobrojnih promenljivih kao i „proširenih adresa" u nekim potprogramima operativnog sistema koji predviđaju postojanje drugog procesora (OSWORD, OSFILE i slični). Jasno je da EQUW može uvek da se zameni sa dve, a EQUD sa četiri naredbe EQUB ali je Acorn, očigledno, želeo da svojim korisnicima pruži pun komfor.

EQU naredbe bejzika 2 imaju još jednu lepu osobinu: njihov argument ne mora da bude broj ili string nego bilo kakav izraz koji bejzik interpretator može da izračuna. To, samim tim, znači da nam je dozvoljeno da pozivamo funkcijske potprograme koji nas „izvode" iz asemblera a zatim vraćaju u njega. Na ovaj način moguće je koristiti BBC-jev asembler kao tzv. makro asembler.

```
LIST
   10 MODE 0:VDU 28,0,31,39,16,
   20 VDU 24,0;512;1273;1023;
   30 VDU 29,0;512;:VDU 19,1,0;0;
   40 VDU 19,0,3;0;:MOVE 0,0
   50 DRAW 400,400

   10 OSWRCH=&FFEE
   20 DIM START 500
   30 FOR I%=0 TO 3 STEP 3:P%=START:RESTORE
   40 [OPT I%
   50 LDA #0
   60 STA &70
   70 STA &71
   80 EQUS FNVDUK(39)
   90 .KRAJ
  100 LDX &70
  110 RTS:]:NEXT I%
  120 CALL START:END
  130 DATA 22,0,28,0,31,39,16,24,0,0,512,1273,1023,29,
      0,0,512,19,1,0,0,0,0,19,0,
3,0,0 0,25,4,0,0,0,0,25,5,400,400
  140 DEFFNVDUK(N):LOCAL D,D$,H,L,J%
  150 FOR J%=1 TO N:READ D$:D=EVAL(D$)
  160 IF D>255 THEN H=D DIV 256:D=D MOD 256 ELSE H=-1
  170 IF D<>L THEN [OPT I%:LDA #D:]
  180 [OPT I%:JSR OSWRCH:]
  190 L=D:IF H<>-1 THEN D=H:H=-1:GOTO 170
  200 NEXT J%
  210 =""
```

Na slici 1 prikazan je kratak bejzik program koji crta crnu liniju u gornjoj polovini ekrana dok je pozadina žuta. Da bismo ovaj program napisali na mašinskom jeziku, potrebno je da ga pretvorimo u jednu jedinu VDU naredbu čiji bi početak izgledao otprilike ovako: VDU 22,0,28,0,31,39,16, . . .

Najjednostavniji način za prelazak na mašinski kod bi se sastojao od sastavljanja tablice (uz korišćenje naredbe EQUB ili, prema prirodi stvari, EQUW) iz koje bi se učitavali podaci i prosleđivali potprogramu OSWRCH kao parametri. Na slici 2 je, međutim, prikazano jedno neracionalnije ali daleko duhovitije rešenje, koje se zasniva na kreiranju makro naredbi.

```
   10 OSWRCH=&FFEE
   20 DIM START 500
   30 FOR I%=0 TO 3 STEP 3:P%=START:REST
ORE
   40 [OPT I%
   50 LDA #0
   60 STA &70
   70 STA &71
   80 EQUS FNVDUK(39)
   90 .KRAJ
  100 LDX &70
  110 RTS:]:NEXT I%
  120 CALL START:END
  130 DATA 22,0,28,0,31,39,16,24,0,0,512
,1273,1023,29,0,0,512,19,1,0,0,0,0,19,0
3,0,0,0,25,4)0,0,0,0,25,5,400,400
  140 DEFFNVDUK(N):LOCAL D,D$,H,L,J%
  150 FOR J%=1 TO N:READ D$:D=EVAL(D$)
  160 IF D>255 THEN H=D DIV 256:D=D MOD
256 ELSE H=-1
  170 IF D<>L THEN [OPT I%:LDA #D:]
  180 [OPT I%:JSR OSWRCH:]
  190 L=D:IF H<>-1 THEN D=H:H=-1:GOTO 17
0
  200 NEXT J%
  210 =""
```

Program 2 daje isti rezultat kao da ste otkucali:

```
LDA #22
JSR OSWRCH
LDA #0
JSR OSWRCH
LDA #28
JSR OSWRCH i tako dalje.
```

Jasno je da niko pametan ne bi kucao 78 praktično jednakih naredbi, pa se vredi potruditi da se razume način na koji je kompjuter naveden da ih otkuca sam. Čitav posao je obavila linija 80, koja je pozvala potprogram FNYDU. Ovaj potprogram najpre učitava podatke iz DATA lista i smešta ih u promenljivu D, koristeći D=EYAL D$ (smisao ovog indirektnog učitavanja je da se dopusti da elementi DATA liste budu heksadekadni brojevi). Linija 160 proverava da li je D veće od 255 i u slučaju potvrdnog odgovora deli ovu promenljivu na viši i niži bajt A i D. Posle ove konverzije, fleg H dobija vrednost −1 da bi računar znao da treba da pošalje dva broja sistemskom potprogramu. Linija 170 predstavlja nešto što bi moglo da se nazove uslovno asembliranje i doprinosi optimizaciji programa. Ukoliko je kod karaktera koji želimo da prikažemo jednak kodu prethodnog prikazanog karaktera, nemamo potrebe da ponovo smeštamo nešto u akumulator — stari sadržaj je očuvan. Ukoliko je promena sadržaja neophodna, u mašinski program

U „Galaksiji" 150 smo govorili o nekim korisnim ali neobjašnjenim naredbama i opcijama koje stoje na raspolaganju vlasnicima operativnog sistema 1.20. Tada smo obećali da ćemo se u „Računarima 3 pozabaviti daleko plodnijim poljem — neobjašnjenim naredbama bejzika 2. One za ambicioznijeg programera, posebno ako programira u mašinskom jeziku, predstavljaju pravu premiju koja se dobija samo jednom u (razume se, računarskom) životu.

se ubacuje naredba LDA#nn. Iza srednje zagrade koja označava početak asemblera uvek se nalazi komanda OPT, ako se ona izostavi, BBC podrazumeva OPT 3, što, istina, u ovom slučaju ne bi izazvalo nikakve loše rezultate.

Linija 190 priprema promenljivu L koja (eventualno) sadrži drugi bajt koji treba proslediti potprogramu OSWRCH. Pošto, na već opisan način, argument bude prosleđen, nastavlja se FOR-NEXT petlja koja učitava sve „stavke" DATA liste iz linije 130, a zatim se glavnom programu vraća prazan string (linija 210). Kakvoga ima smisla vraćati prazan string? Setimo se da smo u glavnom programu koji se asemblira izvršili EQUS FNYDU(39) da bi privremeno „napustili asembler". Iz funkcije mora da se vrati neka vrednost koja je samo formalne prirode. Da smo vratili, na primer, slovo A, ono bi bilo ubačeno u tekst i docnije, pri izvršavanju programa, bilo interpretirano kao mašinska instrukcija EOR sa neočekivanim posledicama.

Osim EQU komandi, bejzik 2 proširuje asembler jednom veoma zgodnom opcijom: relociranim upisom programa. Potrebno nam je, na primer, da opremimo tekst procesor YIEW drajverom za štampač. Iz dokumentacije ROM-a saznajemo da printer drajver mora da bude napisan u veoma preciznom formatu i da počinje od &400. Ukoliko napišemo:

```
10 FOR I%=0 TO 3 STEP 3
20 P%=&400
30 [ OPT I%
40 ! mašinski program
998 ]
999 NEXT I%
```

a zatim izvršimo RUN, doći će do totalne konfuzije. Šta se dogodilo? Sadržaji celobrojnih promenljivih A%–Z% su smešteni negde na četvrtoj strani memorije (&400–&4FF). Računar je u ovaj prostor počeo da upisuje mašinski program dok nije upisao besmislen sadržaj u promenljivu P%, koja sadrži adresu tekuće instrukcije koja se asemblira. Promenom ove vrednosti mašinski program počinje da se upisuje na neko slučajno mesto, što izaziva nepredvidljive posledice.

Kako da se reši ovaj problem? Program možemo da upišemo u neko slobodno područje memorije, snimimo ga na kasetu i prenesemo na četvrtu stranu. Ništa lakše: umesto P%=&400 možemo, na primer, da upotrebimo P%=&4000! No, da li je baš tako? Kada na ovaj način asemblirani program „pomerimo" na &400 nastaje totalni krah, jer JMP i JSR naredbe uopšte ne rade! Ove naredbe sadrže apsolutne adrese, pa nije svejedno da li je asembler podrazumevao „oridžin" &400 ili &4000. Potrebno nam je, dakle, da računar asemblira program kao da se nalazi na četvrtoj strani memorije, a da kod smešta na stranu &40. Ovo omogućuju nove naredbe bejzika 2:

```
10 FOR I%=0 TO 7 STEP 7
20 P%=&400
30 O%=&4000
40 [ OPT I%
50 ! mašinski program
999 NEXT I%
```

Uputstvo za upotrebu nas, na stranama 447—448, uči da koristimo opcije 0–3 asemblera. OPT 4 je, kao što svakako pretpostavljate, „specijalitet" vašeg bejzika. Upotreba ove opcije je očigledna iz gornjeg primera: program se upisuje u područje memorije koje pokazuje promenljiva O%, dok je P%, kao i ranije, logički oridžin programa. OPT 7 je kombinacija OPT 3 i OPT 4.

Šta da radimo sa ovako napisanim programom? Snimićemo ga na kasetu ili disk kucajući *SAVE „DRAJVER" 4000 +100 400 400 koristeći, tako, nedokumentovanu opciju operativnog sistema 1.20 koja je objašnjena u „Galaksiji": segment memoriji &4000–&4100 će biti snimljen tako da se docnije učitava počevši od &400.

. . . nove veze bejzika i mašinca . . .

Šta bismo radili da drajver koji smo upravo pisali nije fiksne dužine (&100 bajtova)? Jedno rešenje bi bilo da pogledamo njegov asemblerski listing, zapišemo adresu poslednje instrukcije, a zatim otkucamo *SAVE, navodeći tu adresu kao drugi argument. Iako je rešenje sasvim praktično, teško da može da se označi kao elegantno, na stranu činjenica da nikada ne bismo mogli da ga primenimo u nekom komercijalnom programu koji bi, na primer, generisao drajver za štampač prema zahtevima korisnika. Potrebno nam je, dakle, da nekako izvršimo naredbu *SAVE, navodeći promenljive a ne konstante kao njene argumente.

Problem je u tome što *SAVE (kao i sve naredbe koje počinju zvezdicom) nije bejzik naredba već komanda koju obrađuje operativni sistem. Operativni sistem, izgleda, ne želi da ima mnogo posla sa bejzikom, pa ne priznaje u njemu definisane promenljive. Pogled na 463. stranu „Uputstva" govori da se iz bejzika može proslediti komanda operativnom sistemu primenom sistemskog potprograma CLI koji počinje od &FFF7. Ovakvo prenošenje je, međutim, komplikovano, jer zahteva formiranje kontrolnog bloka negde u memoriji, promenu sadržaja procesorskih registara i neke druge sitnice. Zato su konstruktori bejzika 2 predvideli naredbu OSCLI, koju ćemo najbolje upoznati ako otkucamo OSCLI „MOTOR 1". LED dioda „cassette motor" se, primećujemo, upalila što znači da OSCLI „MOTOR 1" deluje potpuno isto kao *MOTOR 1!

Snimanje bismo, dakle, mogli da obavimo sa OSCLI „SAVE 4000„+STR$(Q%)+" 400 400" da nije jedne sitnice: STR$(Q%), doduše, pretvara sadržaj promenljive Q% u string, ali je taj string napisan kao niz dekadnih cifara, dok naredba *SAVE očekuje heksadekadne adrese! Problem izgleda nerešivo (tako je mislio i autor ovoga teksta kada je pripremao program „Potraga" koji je objavljen u „Galaksiji" 148, problem je tada rešen pripremom jednog podužeg mašinskog programa koji je pozivao rutinu operativnog sistema OSFILE), ali bejzik 2 nudi sasvim jednostavnu soluciju: OSCLI „SAVE 4000" „+STR$(Q%)+" 400 400". Primetimo da tilda ispred zagrade izaziva prevođenje sadržaja promenljive Q% u heksadekadno iskazan string!

. . . i dodaci bejzika

Konstruktori bejzika 2 su, očigledno, smatrali da je bejzik 1 toliko dobro urađen da ga nema potrebe dopunjavati. Odlučili su se, ipak, da otklone jednu nekonsistentnost naredbi za rad sa datotekama i da prošire naredbu PLOT.

Naredba OPENOUT kreira datoteku pripremljenu za upis podataka iz memorije računara. OPENIN je u bejziku 1 otvaralo datoteku kako za čitanje tako i za upis podataka: ovakav pristup ima jedino smisla kada se datoteka nalazi na disketi. U bejziku 2 OPENIN otvara datoteku iz koje će se podaci isključivo čitati, dok OPENUP otvara datoteku čiji će sadržaj biti čitan i menjan (na engleskom se ovakav pristup naziva „update").

Naredbe PLOT 72—79 su u okviru „Uputstva", označene kao 'Reserved'. Ova rezervacija je već iskorišćena: bejzik 2 (uz saradnju OS 1.20) koristi ove opcije kao pomoć pri ispunjavanju većih površina nekom bojom. Svaka od ovih PLOT komandi pretražuje tačke levo i desno od one koja je navedena kao argument sve dok sa svake strane ne pronađe po jednu tačku koja nije u boji pozadine. Zatim će biti izvršen MOVE u levu tačku i DRAW do desne, što rezultira horizontalnom linijom koja prolazi kroz navedenu tačku. Slično tome, naredbe PLOT 88—95 traže prvu tačku desno od navedene koja je nacrtana u boji pozadine, a zatim vrše MOVE u tu tačku.

Izgleda da i mnoge druge PLOT komande imaju „specijalne funkcije", ali autoru ovoga teksta nije uspelo da ih pronađe čak ni uz pomoć literature. Pokušajte! Možda ćemo rezultate vašeg istraživanja objaviti u „Računarima 4"!

Dejan Ristanović

Biblioteka programa ZX SPECTRUM

SPEKTRUM SA DESET RUKU

Uobičajen način rada sa računarom, na koji smo svi navikli i koji nam se zbog toga čini kao jedino moguć, jeste upotreba jednog programa — onog koji se trenutno nalazi u memoriji. Međutim, često se dešava da taj program ne zauzima ni polovinu raspoloživog memorijskog prosotra. Savremene računare odlikuje mogućnost rada sa više programa u isto vreme. Ovom pogodnošću se, uz malo veštine, mogu opremiti i popularni „spektrumi". Multiprogramski rad je, naravno, najmanje pitanje mode. On može značajno da unapredi primenu računara.

Nikome, naravno ne smeta kada memorija „zvrji prazna". Problem bi bio tek kada je ne bi bilo dovoljno. Pa ipak, mora da ste razmišljali o tome kako da iskoristite višak prostora. Recimo da imate nekoliko kraćih programa, a desi se situacija da treba naizmenično da ih koristite. Tada ćete, po svoj prilici, najveći deo vremena da izgubite petljajući sa kasetofonom. Zar ne bi bilo jednostavnije kada bi svi programi mogli da se ubace u računar, a onda prosto pozivāju jedan po jedan, kada na njih dođe red!

Multiprogramski rad je lako izvesti ako programi nemaju zajedničkih linijskih brojeva i ako ne koriste ista imena za promenljive. Jednostavno se od svih programa oformi jedan jedini, korišćenjem, na primer, naredbe MERGE. Ali šta uraditi ako programi koriste iste programske linije i iste promenljive? Menjati linijske brojeve i varijable uopšte nije privlačna ideja, posebno ako programi i nisu baš sastavljeni od dve ili tri linije.

Unošenje u računar

Očigledno, vredi potražiti način da se više programa istovremeno drži u memoriji. Ako ste, međutim, tek nabavili računar, onda vam sve ovo, verovatno, i ne zvuči preterano ubedljivo — vi ste još uvek u fazi akcionih i drugih igara. Ali čak i u tom slučaju, nemojte da preskočite ovaj tekst. Možda imate dve-tri igre pisane na bejziku, a prosto ne znate koja vam se više sviđa. Najjednostavnije rešenje je da ih sve ubacite u računar, a onda igrate neizmenično. Ne morate se plašiti da vaše malo programsko iskustvo nije dovoljno za posao koji vas ovde očekuje. Sve što treba da uradite jeste da, sledeći naša uputstva, ukucate programe koje dajemo.

Englez Mark Džons je nedavno izneo jednu dosta interesantnu ideju za multiprogramski rad na „spektrumu". Mi smo tu ideju malo modifikovali, uvodeći neka poboljšanja i proširenja. Tako sada postoji i jednostavan način za prenošenje promenljivih iz jednog programa u drugi.

PROGRAM A

```
  10 PRINT "Izaberite program (1
-";PEEK 26142;")"''"Pritisnite E
NTER za otvaranje   novih progra
ma"
  20 IF INKEY$<>"" THEN GO TO 20
  30 LET k$=INKEY$
  40 IF k$=CHR$ 13 THEN GO TO 80
  50 IF k$<"1" OR k$>STR$ PEEK 2
6142 THEN GO TO 30
  60 CLS : POKE 25824,VAL k$
  70 RANDOMIZE USR 25822
  80 CLS : INPUT "Ukupan broj pr
ograma (1-9) ? ";n
  90 IF n<1 OR n>9 THEN GO TO 80
 100 LET rp=26500
 110 LET rtop=PEEK 23732+256*PEE
K 23733-21*8: LET b=rtop-rp
 120 DIM a(n): FOR f=1 TO n
 130 PRINT AT 21,0;"PROGRAM ";f;
 140 INPUT "Preostala memorija:
";(b)''"Broj bajtova predvidjeni
h"'"za program ";(f);" ? ";y

 150>IF y<200 THEN PRINT #0;"200
 BAJTOVA JE MINIMUM !": PAUSE 10
0: GO TO 140
 160 LET b=b-y: IF b-(n-f)*200<0
 THEN PRINT #0;"NEMA DOVOLJNO PR
OSTORA ZA OSTALEPROGRAME !": PAU
SE 100: LET b=b+y: GO TO 140
 170 PRINT AT 18,9;" : ";y;" baj
tova"
 180 LET a(f)=y: NEXT f
 190 PRINT #0;"PREOSTALA MEMORIJ
A : ";b'"(startna adresa : ";rto
p-b;")"'',"pritisnite ENTER": PA
USE 0
 200 LET y=25500: FOR f=1 TO n
 210 LET y=y+12: POKE y+1,INT (r
p/256): POKE y,rp-256*PEEK (y+1)
 220 LET rp=rp+a(f): NEXT f
 230 LET y=y+12: POKE y+1,INT (r
p/256): POKE y,rp-256*PEEK (y+1)
 240 POKE 26142,n
 250 RANDOMIZE USR 26140

 260 RUN
1000 CLEAR 25499: LOAD ""CODE 25
705,590: RUN 80
2000 SAVE "multi-prog" LINE 1000
```

Da biste videli kako sve to izgleda, ukucajte prvo priloženi program A. Kada ste se uverili da niste nigde pogrešili naročito ne u adresama uz POKE, PEEK i USR, startujte program sa RUN 2000, čime ćete obaviti snimanje na kasetu. Posle toga možete izbrisati celu memoriju i ukucati program B. Sada ne morate previše brinuti ako i pogrešite u nekoj od DATA linija. Prosto mirno otkucajte ceo program i startujte ga sa RUN. Izvršavanje traje nešto više od pola minuta. Program čita svaku DATA liniju i na osnovu nje formira mašinski kod, počev od adrese 30000. Pri tome se proverava zbir kodova svih karaktera u DATA listi, i ako se otkrije greška, bićete o tome obavešteni, uz broj linije gde je greška detektovana. Proverite listu, ispravite grešku i ponovo startujte program. Kada sve bude u redu, mašinski kod će automatski biti poslat na kasetu. Snimite ga odmah iza programa A i time je vaš posao oko pravljenja programa završen. A njegovo korišćenje je još jednostavnije: čim ga budete uneli u računar pomoću naredbe LOAD „", program će se automatski startovati i davati vam dalja uputstva. Prvo se od vas očekuje da zadate ukupan broj programa sa kojima

PROGRAM B

```
   5 CLEAR 29999
  10 DATA "fff3f5c5d5e52a595c7ef
e2a206d3620cd4f65cd696511a066cd9
c653a1e66c630cda66511b466cd9c65c
d9e02cd1e03fe41"
  11 DATA 7170
  20 DATA "ca3e65d63038f1473a1e6
6b838ea3ab05cb82819c5fdcb47cecdc
765f132b05cfdcb478ecdc7653ab05ca
7285ecdba652100"
  21 DATA 7295
  30 DATA "0022495ccd4f65cd6965c
d1166fbc3a212f33e00f518ccfe2b203
9237ed630383247 3a1e66b8382b7832b
15c36c02b36f93e"
  31 DATA 6717
  40 DATA "32cd810f3e36cd810f3e3
2cd810f3e32cd810f3e38cd810f3e0d3
2085cfdcb01eee1d1c1f1fbc9cdb0163
ef7cd810fcdba65"
  41 DATA 7414
  50 DATA "cd4f65cd6965cd1166fbc

2b412cdba65cd4f652a4f5cfdcb019ef
bc338123a485ce6380f0f0ff60806012
e0010fed3feee10"
  51 DATA 7284
  60 DATA "06802d20f5c9972100400
1ff1877230bb820fafd5653060372230
bb820fa119766cd90653ab05cc630cda
665c92101402293"
  61 DATA 6347
  70 DATA "662120182295661afeffc
8cda6651318f6d52a9366ed4b9566cd2
40b229366ed439566d1c9d12ab25c2bf
92b2b223d5cd5c9"
  71 DATA 6840
  80 DATA "cd0066114f5ccdef65115
75ccdef6511535ccdef65114b5ccdef6
511595ccdef6511b25ccdef65c901020
0fdcb474e2003ed"
  81 DATA 7235
  90 DATA "b0c9ebedb0ebc93ab05cc
b27174717805f1600219c6319c921ff0
33d20fd2bbc20f9c9f30609c578cd036

65e235623e521af"
  91 DATA 6901
 100 DATA "15011500edb0e11bcd696
613cd6966cd69663e801213cd69663e0
d12133e801223235e23562b2b2b1b3e3
e12cd6966c110c2"
 101 DATA 6492
 110 DATA "9732b05c3e14ed47ed5ef
bc973237223c9ed56c9ed5ec93ab15cc
d036601060009 5e2356237e23666fed5
2444dd5f7e1d5ed"
 111 DATA 6938
 120 DATA "b0c3cd0900000000050726
f6772616d20ff3a204e45573f202f205
0726f6772616d28302dff293fff"
 121 DATA 5087
1000 LET a=30000: LET u=0
1010 FOR n=1 TO 12: PRINT n*10:
LET s=0: READ a$
1020 FOR h=1 TO LEN a$ STEP 2
1030 LET p=CODE a$(h): LET q=COD
E a$(h+1)

1040 LET s=s+p+q
1050 IF p>96 THEN LET p=p-39
1060 IF q>96 THEN LET q=q-39
1070 LET p=p-48: LET q=q-48: POK
E a,16*p+q
1080 LET u=u+PEEK a
1090 LET a=a+1: NEXT h
1100 READ t: IF t<>s THEN PRINT
"DATA ERROR !": STOP
1110 NEXT n: PRINT "DATA OK"
1120 IF u<>66930 OR a<>30590 THE
N PRINT "POKING ERROR !": STOP
1130 PRINT "POKING OK": SAVE "co
de"CODE 30000,590
```

ćete raditi, a zatim treba svakom programu dodeliti njegov memorijski prostor. Konačno, izabraćete jedan od programa da biste u njega ušli. Od tog trenutka, svi ostali programi kao da iščezavaju, sve dok ne budete poželeli da uđete u neki od njih.

Multiprogramske tehnike

Tehnika kojom se postiže rad sa većim brojem programa je prilično prosta, ali mi nećemo moći ovde da je detaljnije izložimo. Ako vas to, ipak, neizostavno interesuje, moraćete malo da se pomučite oko analiziranja programa. Inače, ideja je u tome da svaki bejzik program čuva svojih šest sistemskih promenljivih: CHANS, DATADD, PROG, E_LINE i RAMTOP, koje se, kada na to dođe red, smeštaju na svoja uobičajena mesta među sistemskim varijablama. Naravno, taj posao ne obavljamo mi, već posebna mašinska rutina. Naše je samo da obavestimo procesor o tome da želimo promenu programa. A to vršimo jednostavnim ukucavanjem zvezdice (CHR$ 42) u prvi karakter „nulte" programske linije, na isti način kao što bismo otkucali bilo koju drugu direktnu naredbu.

Procesor je postavljen u „interapt mod 2", a to znači da otprilike svakih 50 ms dolazi do prekida redovnog programa i prelaza na izvršenje posebne „interapt rutine". Ona se, u našem slučaju, nalazi na adresi 25705 i tu se prvo testira sadržaj prostora za editovanje (E_LINE). Ako se naiđe na zvezdicu, to je signal da treba izvršiti promenu programa. Od nas se tada očekuje da izaberemo program u koji hoćemo da uđemo. Druga mogućnost je da pritisnemo NEW, čime brišemo tekući program i ponovo se u njega vraćamo, dok ostali programi pri tome ostaju nedirnuti.

Vodite računa da naredbu NEW ne upotrebite u nekoj drugoj prilici. Ako imate naviku da je svaki čas koristite, počnite da se odvikavate.

Maksimalan broj programa sa kojima možete raditi je deset, i svaki je označen jednim brojem između Ø i 9. Program Ø ne možete koristiti za svoje potrebe. Ona služi za formiranje ostalih programskih blokova. Pomoću njega birate ukupan broj programa sa kojima ćete raditi i određujete memorijski prostor za svaki od njih. Imajte u vidu da pri obrazovanju novog seta programa automatski brišete čitav prethodni set. Program Ø možete pozvati kao i bilo koji drugi, kada god to želite. Ako to, međutim, uradite slučajnim neželjenim pritiskom na nulu umesto na pravu cifru, ne morate se uzbuđivati. Biće vam pružena još jedna prilika za izbor programa.

Programski set

Kada formirate programski set, možete organizovati memoriju i tako da vam preostanu slobodni bajtovi za vaše mašinske rutine i slično. Čim unesete veličinu memorijskog bloka za poslednji program, dobićete informaciju o dužini i početnoj adresi slobodnog prostora. On ide od RAMTOP-a poslednjeg programa, pa sve do grafičkih znakova, koji su, inače, zajednički za sve programe.

Svaku promenu programa prati karakterističan zvučni signal. Ako vam se visina tog tona ne sviđa, promenite sadržaj adrese 25956. Tamo se, inače, nalazi broj 128. Probajte, recimo, POKE 25956,64. A postoji i lak način da se tog zvuka zauvek oslobodite: otkucajte samo POKE 25954,Ø. Zvuk se ponovo uključuje sa POKE 25954,16.

Ako hoćete da upotrebite printer, vratite procesor u mod 1, pomoću RANDOMIZE USR 26222. Mod 2 se ponovo postavlja sa RANDOMIZE USR 26225.

Kada želite da uzmete promenljive iz nekog drugog programa, jednostavno otkucajte znak „+" i iza njega broj programa iz koga se promenljive uzimaju. Tako na primer „naredba" +2 znači da će promenljive iz programa 2 biti prenete u tekući program. Inače, ovo prenošenje promenljivih radi na istom principu kao i naredba MERGE, samo što se ne čitaju podaci sa kasete već neposredno iz memorije računara.

Konačno, ako se budete odlučili da disasemblirate mašinski program, biće vam potrebne sledeće informacije:

Mada je mašinski program formiran na adresi 3ØØØØ, njega treba uneti u memoriju počev od 25705. Čitav kod zauzima 590 bajtova, ali sam program se praktično završava na 26258. Preostalih 36 bajtova do 26294 su tabele sa tekstom i privremenim memorijskim lokacijama.

Koristiće vam i spisak važnijih rutina i potprograma:

25705	Interapt rutina. Testira se E_LINE, skanira tastatura i izvršava odgovarajuća akcija.
25822	Rutina za promenu programa pri dolasku iz programa 0.
25935	Potprogram za generisanje zvuka.
25961	Brisanje ekrana i ispisivanje tekućeg programa.
26000	Štampanje teksta u vrhu ekrana.
26012	Štampanje u produžetku nekog prethodnog teksta.
26022	Štampanje znakova.
26042	Brisanje mašinskog steka novoizabranog programa.
26055	Zamena sistemskih varijabli.
26112	Nalaženje adrese sistemskih varijabli datog programa.
26192	Vremenska pauza.
26140	Rutina za inicijaiizaciju sistema pri formiranju novog seta programa.
26228	Rutina za prenošenje bejzik varijabli. Interpretacija naredbe „+".

Jovan Skuljan

Biblioteka programa BBC B
KAKO PRILAGODITI „VIEW"

Značajan broj vlasnika BBC računara odabira ovaj kompjuter zbog izvanrednih mogućnosti za obradu teksta. Za takvu obradu je neophodno posedovati odličan program kao što je „View", „Wordwise" ili „Merlin Scribe". Može, međutim, lako da se desi da potrošite nekih 900 funti (po 400 za računar i štampač i 70 za ROM sa programom) da biste otkrili da u tekstu koji obrađujete ne mogu da se nađu naša slova č, ć, ž i š. Znači li to da ste napravili tešku grešku? Ne, pogotovu ako imate malo programerske veštine i ako ste se opredelili za Ejkornov tekst procesor „View".

Definisanje naših slova je problem koji mora da se reši u dve faze. U okviru prve faze morate da definišete oblike slova koji će se prikazivati na ekranu i da ih dodelite pojedinim tasterima. U drugoj fazi treba da sastavite printer drajver, to jest program koji će prilagođavati tekst koji računar šalje obliku koji štampač „razume". Prva faza je nezavisna od štampača koji posedujete i možete sa njom završiti kucajući program koji dajemo. Što se druge faze tiče, moraćete da propratite naš primer i pročitate objašnjenja i sve to prilagodite sopstvenom printeru.

Slova se na ekranu definišu primenom naredbe VDU 23, na način koji je potpuno objašnjen u okviru uputstva za upotrebu. Postoji, međutim, nešto što tamo nije objašnjeno ili, bar, nije objašnjeno u dovoljnoj meri. Da biste dodelili naša slova tasterima, potrebno je da redefinišete neke kodove. Jedno od mogućih rešenja, izabrano tako da što više podseća na raspored na pisaćoj mašini, moglo bi da bude:

slovo	taster
č	
ć	
ž	
š	

Potrebno je, dakle, da predefinišemo znake čiji su kodovi redom 91, 95, 93 i 92, odnosno 123, 96, 125 i 124 kada se radi o velikim slovima. Sledeći uputstvo moraćemo, dakle, da izvršimo *FX 20,4, čime će kilobajt ionako prilično „stanjene" memorije biti potrošen na predefinisanje svega 8 slova. Ovaj gubitak postaje još značajniji kada radimo sa disk jedinicama.

Pokazalo se da je vredelo utrošiti vreme na razumevanje BBC-jevog mehanizma definisanja znakova. Pokazuje se da memorijske lokacije &368 — &36E sadrže informacije o stranama memorije u kojima se nalaze definicije znakova čiji su kodovi &20-&3F, &40-&5F itd. U početku se u svim ovim lokacijama nalazi &C pošto je strana &C(&C00—&FF) rezervisana za definisanje znakova. Za nas je dovoljno da u memorijsku ćeliju &369 upišemo &B, pa će se definicije za velika slova upisivati počevši od &B00, tj. u prostor koji normalno koriste definicije narandžastih tastera koji, dok radimo sa programom „View" i onako imaju sasvim specijalne funkcije. Uz ovaj trik utrošak korisnog memorijskog prostora na definisanje karaktera iznosi nula bajta — retka prilika u programiranju kada bolje rešenje ne postoji!

Kako je najlakše realizovati definisanje karaktera? Sastavićemo program koji se učitava sa kasete ili diska, definiše karaktere i, na kraju inicijalizuje sam „View". Ovaj program bi bilo lako i lepo napisati na bejziku ali — zašto da malo ne vežbamo asembler?

Kada unesemo i isprobamo program, počinju da nam rastu apetiti. Zašto da se posle svakog stratovanja „View"-a „mučimo" pritiskajući 'CAPS LOCK' i prelazeći na mala slova? Zašto da kucamo NEW, a zatim MODE 6, da bismo počeli da pišemo? Zašto, najzad, da kucamo PRINTER EPSON ili PRINTER SEI da učitamo drajver za naš štampač? Sve ove poslove će, umesto nas, obaviti program.

Nastavak na 49. strani

S KNJIGOM ĆE SVE BITI LAKŠE

PRIRUČNICI, PERIODIKA, KASETE, ZA SVE ZALJUBLJENIKE U RAČUNARE, IGRAČKE NAŠEG DOBA — KOJI SU I VIŠE OD TOGA

Najveća slovenačka knjižara — Mladinske knjige u Ljubljani, Titova br. 3 — nudi vam, na svom posebnom računarskom odeljenju, najbogatiji i najsvestraniji izbor stručne i popularne literature s područja računarstva i informatike:

Literatura na slovenačkom jeziku:

1.	Ellershaw, Schofield: BASIC za začetnike	400 din
2.	Špiler: BASIC	1000 din
3.	Gerlič: ABC računalništva (prošireno izdanje)	500 din
4.	Više autora: Osebni računalnik	550 din
5.	Laurie: Čudoviti svet računalnikov (s knjigom BASIC)	2600 din
6.	Više autora: Uvod v računalništvo; Prvi koraki v BASIC-u; Učenje z računalnikom; Grafične in zvočne igre — komplet 4 knjiga	3600 din
7.	Uratnik: Mikroračunalnik — zgradba in arhitektura	625 din
8.	Wirth: Računalniško, programiranje I.	1000 din
9.	Bratkovič: Metode programiranja	714 din
10.	Bratkovič: Programiranje računalnikov	985 din
11.	Bratko: Inteligentni informac. sistemi	680 din
12.	Divjak: Računalniška grafika	476 din
13.	Kodek: Poglavja iz računalniških sistemov	680 din
14.	Kodek: Uvod v mikroprocesorske sisteme	680 din
15.	Bratko, Rajkovič: Računalništvo — PASCAL	671 din
16.	Nadrah: COBOL	900 din
17.	Zakrajšek: FORTRAN	600 din
18.	Friedrich, Schaff: Mikroelektronika in družba	950 din
19.	Hišni računalnik od A do Ž (u pretplati do izlaska)	2800 din

Literatura na srpskohrvatskom jeziku:

20.	Čip, Šahinpašić: Kompjuterska početnica	680 din
21.	Špiler: BASIC (prevod)	950 din
22.	Reljić, Makanac: BASIC	1400 din
23.	Parezanović: BASIC	532 din
24.	Stojković, Tošić: BASIC — zadaci iz programiranja	600 din
25.	Marković: Simbolički progr. jezik ASEMBLER	1076 din
26.	Stojaković, Stojmenović, Tošić: Progr. jezik PASCAL	700 din
27.	Vuletić, Ljubović: Programiranje (FORTRAN)	400 din
28.	Turajlić: Osnove progr. i sistema FORTRAN IV	650 din
29.	Parezanović: Račun. mašine i programiranje FORTRAN IV	450 din
30.	Jauković: Progr. jezik COBOL	350 din
31.	Tkalec: Progr. jezik COBOL	760 din
32.	Stanković: COBOL — zbirka zadataka	650 din
33.	Parezanović: Mašinski i simbolički PDP jezik	504 din
34.	Salčić: Mikroračunarski sistemi	660 din
35.	Grundler: Uvod u mirkroprocesore	600 din
36.	Ribarić: Arhitektura mikroprocesora	1000 din
37.	Sobotka: Mikroprocesori i mikroračunala	1450 din
38.	Đurić: Mini i mirkoračunala	1200 din
39.	Kvaternik: Uvod u operativne sisteme	600 din
40.	Hansen: Principi operativnih sistema	450 din
41.	Petrić: Nelinearno programiranje	1000 din
42.	Vučković: Linearno programiranje	790 din
43.	Čirić: Zbirka rešenih zadataka (Mašinski jezik, Fortran . . .)	500 din
44.	Drajić: Teorija informacija i prenos podataka	420 din
45.	Stajić: Sredstva za obradu podataka	525 din
46.	Altman: Osnovi teorije diskretnog modeliranja	475 din
47.	Krčevinac: Algoritmi i programi iz operac. istraž.	1200 din
48.	Petrić: Operaciona istraživanja — zbirka zadat. I	808 din
49.	Petrić: Operaciona istraživanja — zbirka zadat. II	730 din
50.	Matić: Principi upravljanja komutac. sistemima	860 din
51.	Muftić: Osnovni elementi kompjut. sistema	550 din
52.	Turk, Bodin: Analiza primjenom računala	665 din
53.	Nožica: Analiza primj. računala — primeri i zadaci	1195 din
54.	Žiljak: Simulacija računalom	800 din
55.	Smiljanić: Osnove digitalnih računala	900 din
56.	Smiljanić: Mikroračunala	865 din
57.	Katanić: Uvod u programiranje	1100 din
58.	Aleksić: Računari — organizacija i arhitektura	610 din
59.	Birola: Kompjuterka obrada podataka	400 din
60.	Mrakovčić: Standardizacija programiranja i ERC	900 din
61.	Ferišak: Organizacija EGP	800 din
62.	Arandelović: Kompjut. organizacija u savremenoj privredi	1000 din
63.	Turajlić: Sistem autom. upravlj. — zbirka zadataka	980 din
64.	Matković: Teorija informacija	1400 din
65.	Srića: Sistem informacija — kompjuter	760 din
66.	Ferišak: Osnove informatike	1200 din
67.	Han, Balaban: Osnove informatike	850 din
68.	Više autora: Informacijski sistemi	550 din
69.	Stanković, Tomović: Nelinearni sistemi autom. upravlj.	840 din
70.	Alagić: Principi programiranja	350 din
71.	Alagić: Relacione baze podataka	500 din
72.	Draganović: Adaptivni sistemi upravljanja	450 din
73.	Više autora: Digitalne telekomunikacije	2550 din
74.	Vajzenbaum: Moć računara i ljudski um	350 din
75.	Popović: Elektronski računar — most u budućnost	900 din
76.	Evans: Kompjuterski izazov	800 din
80.	NAJNOVIJE — Laurie: Kompjutor u kući	3000 din
77.	Čavić: Engleski jezik u informatici	665 din
78.	Čavić: Sintakt. osobenosti engl. jezika u informatici	320 din
79.	Više autora: Obrada podataka i programiranje 5-jezični rečnik	3600 din

PORED IZBORA GORE NAVEDENIH KNJIGA MOŽETE KOD NAS NARUČITI ILI DOBITI INFORMACIJE ZA:

— **Računarsku kasetu Radia Študent** (10 originalnih programa s prevodom na srpskohrvatski) za Spectrum 48 K — 1300 din

— **Računar Sinclair ZX Spectrum 16 K i 6 kaseta**
cena s porezom — 69.700,00 din
cena bez poreza — 54.073,00 din

— **Profesionalnu tastaturu za ZX 81 i Spectrum**
cena s porezom — 14.179,00 din
cena bez poreza — 11.000,00 din

— **Profesionalnu tastaturu „Špica" za Spectrum**
cena s porezom — 18.819,40 din
cena bez poreza — 14.600,00 din

— **Dodatni RAM 16 K za ZX 81**
cena s porezom — 22.944,20 din
cena bez poreza — 17.800,00 din

PORUČITE VEĆ DANAS — JER ĆE NAJTRAŽENIJE KNJIGE MOŽDA VEĆ SUTRA BITI RASPRODATE!

Sve knjige kao i računarsku kasetu, računare i pribor možete poručiti pismeno ili telefonom (061/211-895) a poslaćemo vam ih **pouzećem** (platićete poštarinu prilikom prijema paketa). Za pravna lica važi obična narudžbenica. Ispunjenu narudžbenicu pošaljite na našu adresu:

KNJIGARNA MLADINSKE KNJIGE
61000 Ljubljana, Titova 3

mladinska knjiga **Narudžbenica R—3; 11/84**

Potpisani (ime i prezime) .....

tačna adresa .....

neopozivo poručujem (pouzećem) sledeća izdanja:

.....

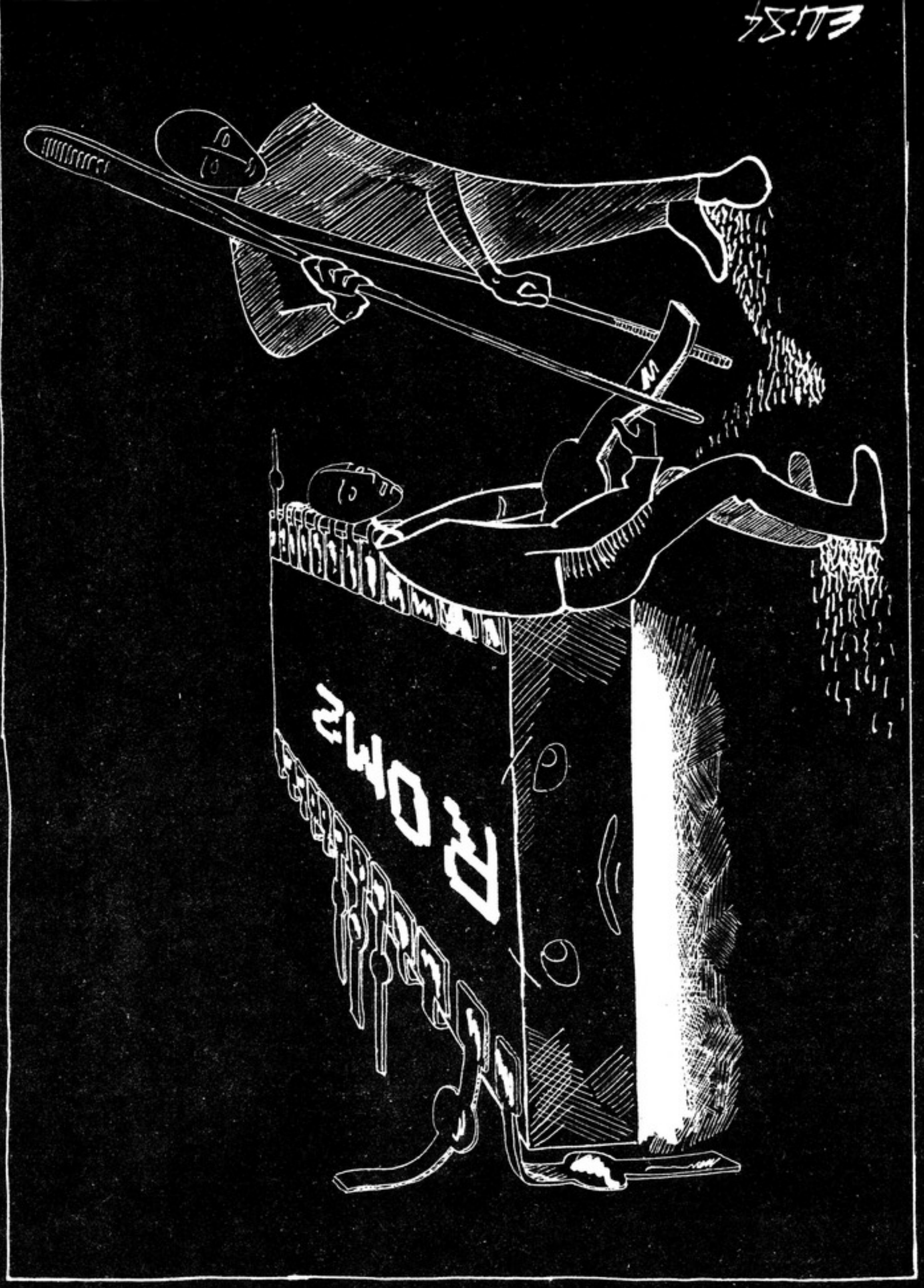
.....

.....

.....

Datum: Potpis:

.....



ROM

računar „galaksija“

Dejan Ristanović
Časopis „Galaksija“
Beograd
Zavod za udžbenike
i nastavna sredstva, Beograd
Elektronika inženjering, Zemun

Sistemski softver računara „galaksija" zauzima, u osnovnoj konfiguraciji, 4 kilobajta memorije. Opremljen njime, računar „galaksija" predstavlja potpuno zaokruženu celinu. Svaki savremen računar je, međutim, koncipiran tako da ga vlasnik, u zavisnosti od potreba i materijalnih mogućnosti, neprekidno proširuje i tako oprema za rešavanje sve složenijih praktičnih problema. ROM 2 je prvo komercijalno softversko proširenje „galaksije". Ono je — u saradnji Zavoda za udžbenike i nastavna sredstva Elektronike inženjering i časopisa „Galaksija" kao i sva ostala proširenja besplatno stavljeno na raspolaganje svim graditeljima ovog već pomalo nacionalnog računara.

ROM 2 predstavlja logičnu nadgradnju osnovnog operativnog sistema. On će „naučiti" vaš računar većini standardnih matematičkih funkcija, proširiti njegov bejzik, podržati rad sa portom za proširenja i štampačem koji je eventualno priključen i omogućiti „galaksiji" da, uz proširenje memorije i hardverske dodatke (emulator i EPROM programator), postane moćan sistem za razvoj softvera za druge mikroprocesorske sklopove koji koriste Z80. Za to je, jasno, posebno značajan kompletan asembler (program za rad sa mašinskim jezikom) koji je ugrađen u ROM 2. One koji planiraju da koriste asembler za razvoj mašinskih programa koji će se izvršavati na samoj „galaksiji" posebno će obradovati činjenica da je omogućeno postavljanje tačaka prekida programa ('break points') i nekoliko drugih monitorskih naredbi.

Opremljena ROM-om 2, „galaksija" ostaje potpuno kompatibilna sa neproširenim modelima. Iako ROM 2, po inicijalizaciji, deluje na neke sistemske promenljive (posebno linkove), sve do sada objavljene informacije o dodavanju naredbi i sličnim „specijalnim efektima" ostaju u važnosti.

Kao što je ROM 2 logična nadgradnja ROM-a 1, tako i ovaj tekst predstavlja nastavak 'Uputstva za upotrebu računara „galaksija"'. On će vam omogućiti da lako koristite mogućnosti dodatka koji vam je upravo stigao iz „Galaksije" i, uz školu mašinca koju objavljujemo u „Računarima" pomoći da sastavite interesantne programe.

INICIJALIZACIJA ROM-a 2
A=USR (&1000)

Da bi iskoristili potencijale ROM-a 2, posle svakog uključivanja računara treba izvršiti tzv. inicijalizaciju — obavestiti računar da novi čip očekuje njegovu pažnju. Posle izvesnog vremena, inicijalizacija će postati toliko prirodna da je nećete ni primećivati. Ukoliko, međutim, baš ne možete da se naviknete na nju, postoji rešenje koje možete da primenite uz malo truda i hardvera: mala izmena ROM-a 1 koja će automatski inicijalizovati ROM 2 posle svakog uključenja „galaksije". Od opreme vam je potreban programator EPROM-a i uređaj za njihovo brisanje. Ako vaša „galaksija" kontroliše EPROM programator, moraćete da pronađete prijatelja koji poseduje „galaksiju" ili da kupite ili pozajmite jedan prazan EPROM 2732.

Najpre otkucajte SAVE 0,&FFF i snimite na kasetu čitav sadržaj ROM-a 1. Zatim sa OLD 12288 relocirano upišite ovaj sadržaj u RAM tako da počinje od &3000 (ili od neke druge adrese na kojoj softver vašeg programatora EPROM-a pretpostavlja da je smešten bafer koji treba prepisati u EPROM). Slede izmene: BYTE &33F9, &C4:WORD &33FA, &1000:BYTE &30/37,29 (time ste, zapravo, smestili CALL &1000 na adrese &3F9-&3FB i promenili bajt &37 u koji je upisana verzija ROM-a 1; do sada je verzija bila 28, a sada je 29.

Posle ovih izmena, isprogramirajte prazan EPROM, zamenite njime ROM 1 i isprobajte računar. Ako je sve u redu, možete da izbrišete vaš bivši ROM 1, isprogramirate u njega sadržaj koji imate na traci i vratite ga u računar vašeg


```
1 !
2 !
3 !      B I O R I T A M
4 !
5 !
6 !    UPUTSTVO  ZA  ROM 2
7 !
8 !
10 HOME
20 P.AT165,"************************"
30 P.AT197,"*   B I O R I T A M   *"
40 P.AT229,"************************"
71 A=KEY(0)
80 ARR$(7)
82 FOR I=0 TO 6:TAKE X$(I):N.I
84 #"SUBOTU","NEDELJU","PONEDELJAK","UTORAK","SREDU","CETVRTAK","PETAK"
100 HOME
110 PRINT "         B I O R I T A M"
120 PRINT "         ==============="
130 PRINT
150 PRINT "OTKUCAJTE DATUM RODJENJA U OB-"
160 PRINT "LIKU <DAN,MESEC,GODINA>."
165 PRINT
170 INPUT X$
180 CALL 3000
210 CALL 1000
220 G=F/7:CALL 2000
230 PRINT:PRINT "RODJENI STE U ";X$(F-G*7);"."
240 PRINT
245 PRINT "OTKUCAJTE DATUM ZA KOJI SE ŽR-"
250 PRINT "TA BIORITAM U ISTOM OBLIKU."
255 PRINT
260 INPUT X$
270 CALL 3000
280 X=F:CALL 1000:F=F-X
285 PRINT
290 PRINT "STARI STE";F;" DANA";:IF F>32740 P.",":PRINT "STO JE PREVISE ZA OVAJ
PROGRAM!":G.999:E.P."."
300 PRINT
310 PRINT "                PRITISNITE RET";
320 IF KEY(0)=13 ELSE G.320
330 HOME
340 FOR I=0 TO 48:DOT 0,I:N.I
350 FOR I=0 TO 64:DOT I,24:N.I
360 TAKE 900
370 FOR J=1 TO 3
380 TAKE A
385 L=0
390 FOR I=F-10 TO F+9
395 FOR K=0 TO 2
400 B=SIND(360*(I+K/3)/A)
401 B=INT(22-B*22+.5)
410 DOT L,B:L=L+1:NEXT K:NEXT I
411 TAKE X:BYTE10271+INT(B/3)*32,X
420 NEXT J
900 #23,70,28,69,33,73
999 A=KEY(0):GOTO 100
1000 F=365*Y+D+31*(M-1)
1010 IF M<3 F=F+INT((Y-1)/4)-INT(.75*(INT(((Y-1)/100)+1))):RET
1020 F=F-INT(.4*M+2.3)+INT(Y/4)-INT(.75*(INT(Y/100)+1)):RET
2000 H=-1
2010 IF G<0 G=32768*H+INT(G+32768):RET:ELSE G=G-32768:H=H+1:GOTO 2010
3000 D=VAL(PTR X$)
3002 I=0
3003 CALL 4000
3010 M=VAL(PTR X$+I)
3011 CALL 4000
3020 Y=VAL(PTR X$+I)
3030 RET
4000 IF BYTE(PTRX$+I)=44 I=I+1:RET:ELSE I=I+1:GOTO 4000
```

ARCTG	1D81
SQR	1D09
POW	1CDA
EXP	1D1B
LN	1B6B
ABS	1BFC
RND na stek	C8F
Poređenje elemenata sa aritm. steka (Z,C)	B10

1
NAREDBE I FUNKCIJE ROM-a 2

naredba	skraćenica	primer
		A=„Z"
%		PRINT %(&2C3A+517)
/		/LABELA
<		početak mašinskog programa
>		kraj mašinskog programa
ABS		PRINT SQR (ABS(A))
		PRINT AB. (100)
ARCTG	AR.	PRINT ARCTG(PI/4)
COS		A=COS(PI/3)
		PRINT COSD(30)
DEL		DEL 100, 200
DUMP	DU.	DUMP &1000,212
EXP		PRINT EXP(1)
INP		X=INP(10)
LDUMP	LD.	LDUMP 0,10
LLIST	LL.	LLIST
		LLIST 1000
LN		PRINT 20*LN(2.367)
LPRINT	LP.	LPRINT „GALAKSIJA"
OUT		OUT 10,&FF
PI		PRINT PI
POW		PRINT POW (8,1/3)
REN		REN 100
SIN		PRINT SIN (2*PI/3)
		PRINT SIND(60)
SQR		PRINT SQR(2)
TG		A=TG(PI/2)
		PRINT TGD(45)

prijatelja. Uključite vaš računar i, dok ste u društvu, prionite na čitanje ostatka ovog uputstva.

Posle svakog uključenja računara, odnosno njegove namerne (PRINT USR(0)) ili slučajne (posle neke vaše grube greške, najčešće pri radu sa asemblerom) reinicijalizacije, otkucajte A=USR(&1000) i pritisnite ENTER. Time je „galaksiji" stavljeno do znanja da je priključen ROM 2 i izvršene neophodne promene područja sistemskih promenljivih. Ukoliko zaboravite ovu inicijalizaciju, neće se dogoditi ništa strašno — normalno ćete raditi sa ROM-om 1 i računar će prijaviti grešku WHAT? kad prvi put pokušate da koristite neku specifičnost ROM-a 2. U tom trenutku možete da otkucate A=USR (&1000) i „galaksija" će raditi sa drugim ROM-om kao da je on aktiviran u samom početku. Neće doći do gubitka programa ili podataka osim promenljive A. Ukoliko želite da sačuvate i nju, upotrebite A=A+0*USR (&1000).

Čitaoci koji su malo zakasnili sa programiranjem ROM-a 1 pa ga šalju redakciji zajedno sa ROM-om 2, dobijaju verziju 29. Oni ove retke mogu da preskoče.

Zagriženije „hakere" će interesovati tačan opis onoga što se dešava u toku inicijalizacije ROM-a 2. „Galaksija", pre svega, briše ekran, što je neophodno za slučaj da je inicijalizacija automatska (u tom slučaju je LD &A, C/RST &20 zamenjeno sa CALL &1000). Zatim se ispravlja bag koji se ispoljava kada je priključeno memorijsko proširenje od 48 Kb; nema, dakle, više potrebe da kucate WORD &2A6A, &FFF0 kad god uključite računar. Posle ovoga, u sistemsku promenljivu 'horizontalna pozicija slike' (&2BA8) upisuje se broj 12; pokazalo se, naime, da je na većini televizora ova promena neophodna; broj 11 odgovara jedino profesionalnim monitorima.

Nakon ispravljanja bagova, menjaju se sadržaji linkova za naredbe (JP &100F) i za video (JF &106F). Ova promena ne bi trebalo da utiče na kompatibilnost propisno napisanog softvera u najgorem slučaju, neki program će isključiti ROM 2 utičući na linkove. Na kraju inicijalizacije ROM-a 2 kontrola se vraća osnovnom operativnom sistemu.

Posle inicijalizacije možete da koristite sve potencijale ROM-a 2 kao da je „galaksija" sa njim rođena. Naredbe se pišu i skraćuju na uobičajeni način. Možete, na primer, da otkucate DUMP &1000,10 i na ekeranu će se pojaviti prvih 10*8=80 bajtova ROM-a 2. Umesto DUMP, odgovara i DU. &1000,10 uz iste efekte.

Pošto „galaksija" opremljena ROM-om 2 ima više naredbi od slova abecede, nije više moguće skratiti svaku od njih na jedno slovo i tačku. Da ste, u gornjem slučaju, otkucali D. &1000, 10, „galaksija" bi izvršila DOT 4096,10, sa neželjenim efektima. Pri prepoznavanju naredbi, dakle, prioritet ima tablica u ROM-u 1, sledi tablica u RAM-u (ako ste je sastavili) i, na kraju, naredbe ROM-a 2. Kompletan spisak naredbi sa primerima i neophodnim skraćenicama dat je na kraju ovoga teksta.

Da bi se olakšala upotreba češće korišćenih funkcija, neki specijalni znaci su dobili posebne uloge. Ukoliko, na primer, poželite da zabavite prste i otkucate /., MNBY, a zatim pritisnete ENTER, „galaksija" neće prijaviti WHAT? Pogodite zašto! Čestitaćemo vam ako uspete u tome, ali mislimo da nećete — odgovor je na sledećih nekoliko stranica.

ROM 2 se isključuje, svako će reći, tako što se odstrani iz računara! Ovakvo isključivanje ne bismo, naravno, mogli da vam savetujemo (ukoliko je ROM 1 promenjen tako da inicijalizuje ROM 2, ovo rešenje je čak i nemoguće) pa predlažemo drugo, čisto programsko: otkucajte BYTE &2BA9,&C9:BYTE &2BAC&C9 i pritisnite ENTER. Time je ROM 2 skoro sasvim isključen. Odakle ovo „skoro"? Ukoliko se u vašem programu nalazi naredba tipa A=USR(&1000), računar u kome se fizički ne nalazi ROM 2 će krahirati, dok kompjuter sa ROM-om 2 neće pretrpeti nikakve posledice. Potpunije isključenje nije moguće.

MATEMATIČKE FUNKCIJE

Računar „galaksija" u osnovnoj verziji ima dobru aritmetiku pokretnog zareza koja je potpuno prilagođena njegovim karakteristikama — šest tačnih cifara je sasvim dovoljno za računar te klase. „Galaksiji", međutim, potpuno nedostaju matematičke funkcije, što za mnoge primene, pogotovu kada se radi o srednjoškolskom obrazovanju, predstavlja vrlo ozbiljan hendikep. Ovaj problem se delimično rešava dodavanjem bejzik potprograma, ali je pravo rešenje tek mašinski rešen set rutina za izračunavanje matematičkih funkcija koji je uklopljen u standardni bejzik. ROM 2 nudi upravo to.

Matematičke funkcije se, naravno, uvek nalaze sa desne strane znaka jednakosti ili iza PRINT naredbe. Iza svake od njih sledi argument koji može da bude konstanta ili brojni izraz; u oba slučaja argument mora da se nalazi u zagradi. Pokušajte, na primer, da otkucate PRINT SQR(4) i pritisnete ENTER. Dobićete 1.99998 — dobrodošli u svet primenjene numeričke analize!

Pre nego što bacite „galaksiju" u koš i vratite se „digitronu" koji ste davno sahranili u fioku, pružite nam još jednu šansu i pročitajte sledeći pasus.

Stepene i eksponencijalne funkcije

PRINT SQR (2)

Trigonometrijske, korene, logaritamske i druge funkcije nikako ne mogu da se izraze kao konačan niz sabiranja, oduzimanja, množenja i deljenja, matematičari su se vekovima mučili da ih izračunaju na ovakav način, sve dok nije dokazano da je ovako nešto nemoguće. Danas koristimo kompjutere koji vrše približno izračunavanje vrednosti ovih funkcija preko beskonačnih redova. Ne biste, sigurno, poželeli da vaša „galaksija" beskonačno dugo radi da bi sabrala beskonačan red, zar ne? „Galaksija" se zato zaustavlja po sabiranju određenog broja članova reda u trenutku kada je tačnost 'zadovoljavajuća'.

Samo, šta je to 'zadovoljavajuće'. Isprobavajući algoritme matematičkih funkcija, morali smo da pravimo kompromise između utrošenog memorijskog prostora, brzine rada i tačnosti. Smatrali smo da je „galaksiji" dovoljno 5 do 6 tačnih cifara; za većinu inženjerskih primena dosta bi bile i dve. Problem, međutim, može da nastupi kada poželimo da dobijemo koren koji bi trebao da bude ceo — i osnovci znaju da je SQR(4)=2 a ne 1.99998. Nema problema — upotrebićemo PRINT INT(SQR(4)+0.5). Ukoliko želimo da zaokružimo rezultat na dve decimale, koristićemo:

PRINT INT (SQR(4)*100+0.5)/100).

Upoznali smo, dakle, funkciju SQR (kvadratni koren) i njena ograničenja. Da nismo malo požurili? Ako ne želimo da već sutra zaboravimo koje funkcije naš računar ima, moraćemo da ih obradimo sistematično i to po grupama.

Svako zna kako se definiše stepenovanje kada je eksponent ceo broj: 2^5 je isto što i 2^2^2^2^2=32. Znamo li šta da radimo kada je eksponent ceo negativan broj? Ništa lakše: 2^−3 je isto što i 1/(2*2*2)=1/8=0.125. Za neke racionalne brojeve problemi su i dalje rešivi: 2^0.5 je isto što i SQR(2) (kvadratni koren). Šta je, međutim, 2^PI ili, još gore, PI^PI?

SISTEMSKE ADRESE ROM-a 2

Ovaj dodatak nije namenjen početnicima u njemu, uz minimum teksta i podrazumevajući značajno predznanje, navodimo informacije koje će vam pomoći da u potpunosti iskoristite vaš ROM 2.

U sledećoj tabeli je data mapa sistemskih promenljivih „galaksijinog" operativnog sistema, bejzik interpretatora i ROM-a 2. Tabela je u mnogome slična onoj koja je objavljena, na strani 26. Uputstva za upotrebu vašeg računara. Pažljiviji čitalac će, međutim, primetiti da su ispravljene dve nebitne greške i dodate nove sistemske promenljive koje koristi ROM 2. Prostor „ispod" bejzika je, dakle, sada potpuno popunjen!

Opis	CALL &
Izraz sa DE — HL	008
Izraz u zagradi DE—HL	A6A
Preskoči blankove	104
Ako je (DE) zarez, kao RST &8 inače WHAT?	5
KEY(0)	CF5
Karakter A na ekran	20
HL na ekran kao ASCII	8FB
Alfanumerik adresiran sa DE na ekran	937
Lociraj varijablu čije ime pokazuje DE. PTR u HL uz C i Z	125
Sledeća naredba	30
Slobodna memorija ARR$-RAMTOP u HL	183
Element izraza sa DE—HL	18D7
Ceo izraz u HL	1866
Pokaži sve registre na ekranu	1978
DUMP od HL za A redova	19DE
Link za naredbe	100F
Link za video	106F
Stanje aritmetičkog steka	8F6
Izraz sa DE na ar. stek	AB2
HL na aritmetički stek	ABC
Četiri bajta adresirana sa HL na arit. stek	A45
Broj sa aritm. steka u HL	A6D
Broj sa aritm. steka u četiri bajta adres. sa HL	73B
Izraz u zagradi sa DE — (IX)	781
+	B32
−	B1E
*	AE6
/	AF7
SIN	1C30
COS	1C24
TAN	1C04

SISTEMSKE ADRESE

adresa	broj bajtova	sadržaj
2800	512	video memorija
2A00	104	promenljive A–Z
2A68	2	pozicija kurzora u memoriji
2A6A	2	kraj memorije (RAMTOP)
2A6C	2	broj HOME zaštićenih bajta video memorije
2A6E	2	izlazni kriterijum FOR-NEXT petlje
2170	16	alfanumerička promenljiva X$
2A80	16	alfanumerička promenljiva Y$
2A91	2	STEP za tekuću petlju
2A93	2	pozicija tekuće linije u FOR-NEXT
2A95	2	bejzik pointer u toku CALL i FOR-NEXT
2A97	2	lokacija od koje treba dampovati
2A99	2	16*ARR$+16
2A9B	2	adresa NEXT promenljive
2A9D	2	pointer za TAKE
2A9F	2	pozicija linije koja se izvršava
2AA1	2	koristi se u toku FOR-NEXT
2AA3	2	privremeni SP (kod CALL)
2AA5	2	diferencijator za tastaturu
2AA7	3	„seed" za RND
2AAA	1	koji prolaz kroz asembler (1. ili 2.)
2AAB	1	Č, Ć, Ž, S postaju C, C, Z, S za bit 7=0
2AAC	124	aritmetički akumulatori *i* stek Z80A
2BA8	1	horizontalna pozicija teksta (inic. &C)
2BA9	3	link za naredbe
2BAC	3	link za video
2BAF	1	sat radi ako je bit 7 setovan
2BB0	1	brojač za pomeranje slike
2BB1	1	fleg za pomeranje slike
2BB2	1	koliko redova treba dampovati
2BB3	1	OPT pri asembliranju
2BB4	1	registar za REPT
2BB5	1	fleg za štampač
2BB6	125	bafer (koristi se pri padu editora, kod bejzik naredbe INPUT i za vreme asembliranja)
2C36	2	početak bejzik programa (početno &2C3A)
2C38	2	kraj bejzik programa

Gde i kako asembler upisuje labele

U trenutku kada asembler završi posao, sve labele bivaju zaboravljene. Ponekad, međutim, može da bude korisno da bejzik program odredi gde se nalazi neki mašinski potprogram označen labelom.

Asembler upisuje labele počevši od kraja memorije. Najpre je upisano ime prve labele kao niz ASCII karaktera, od kojih je poslednjem setovan sedmi bit. Iza (tačnije ispred, jer se sve upisuje od viših adresa prema početku memorije) imena slede dva bajta koja predstavljaju broj dodeljen labeli. Iza toga, bez ikakvog posebnog terminatora, sledi sledeća labela, zatim sledeća... Kraj tabele označen je bajtom nula.

PRINT POW (8,1/3)

Matematički posmatrano, **a^b** je definisano za sve realne brojeve **a** i **b,** pri čemu **a** mora da bude veće od nule ili, eventualno, jednako nuli (tada **b** mora biti različito od nule). To znači da PI^PI može da se izračuna. Ovo računanje se, najčešće, obavlja primenom logaritama. O logaritmima ćemo reći par reči nešto docnije; dovoljno je da znate da vaša „galaksija" sasvim lepo ume da računa **a^b** korišćenjem funkcije POW. POW ima dva argumenta koji se nalaze u zagradi i odvojeni su zarezom. Argumenti su, jasno, **a** i **b,** pa ćete 2^3 izračunati tako što ćete otkucati PRINT POW(2,3) i dobiti 8 ili, da ne budemo prestrogi, bar nešto slično tome.

Posle određenog razmišljanja zaključićete da **a** ne mora uvek da bude negativno da bi a^b bilo definisano. Jasno je, na primer, da je (−2)^3=(−2)*(−2)*(−2)=−8. „Galaksija", međutim, računa funkciju POW korišćenjem logaritama, pa će PRINT POW(−2,3) dati grešku. Da ostvarite celobrojno stepenovanje koje radi u svim mogućim slučajevima, moraćete, dakle, da koristite običnu FOR-NEXT petlju za koju ROM 2 nije ni potreban.

Dok nekom prijatelju sa ponosom pokazujete vaš ROM 2 koji je „galaksiju" naučio matematici, možete da se nađete u velikoj nevolji kada čujete pitanje „a kako se računa kubni koren iz 8?" Ne brinite, „galaksija" ume i to uz malo programerske veštine. Treba da se prisetimo prvog paragrafa ovog poglavlja u kome smo rekli da je SQR(4) isto što i 4^0.5=4^(1/2). Slično tome, i kubni koren iz 8 jednak je S^(1/3), a sedmi koren iz 1200: 1200^(1/7). Znajući da argumenti u POW mogu da budu i izrazi, lako ćemo napisati program poput sledećega:

```
5 ! RAČUNANJE N-TOG KORENA
10 PRINT „KOJI BROJ TREBA DA KORENUJEM" : INPUT A
20 PRINT „KOJI KOREN TREBA DA IZVADIM"; : INPUT B
30 PRINT POW(A, 1/B)
```

Program poput ovoga možemo da koristimo da nađemo kvadratni koren. Zašto je, onda, potrošen prostor u ROM-u za funkciju SQR? Kvadratni koren je u upotrebi mnogo češće od bilo kog drugog, pa ga većina kompjutera odvojeno realizuje. Osim toga, odaćemo vam i jednu malu tajnu: „galaksija" računa SQR(X) kao POW(X,0.5)!

PRINT EXP (1)

„Galaksija" poseduje i funkciju EXP, koja je, za korisnika koji tek ulazi u svet matematike, specijalan slučaj stepenovanja. EXP(X) je, naime, isto što i **e**^×, pri čemu je **e** matematička konstanta barem jednako važna koliko i PI. Broj **e,** poput broja PI, ne može da se izrazi kao konačan decimalni broj, niti kao razlomak; kažemo da se radi o 'transcedentnoj' konstanti. Prvih nekoliko cifara broja **e** su 2.718281828 (može li ovo nekako da se zapamti? Možda i može: u broju **e** se dva puta ponavlja 1828, što je godina Tolstojevog rođenja. Ako znate kada je Tolstoj rođen, znate i broj **e.** Ako, kao većina „hakera", znate broj **e,** možda ćete zaraditi neki poen na času književnosti). EXP(X) je, dakle, isto što i 2.71828^×.

SQR smo nekako objasnili, ali kome je potrebna funkcija EXP kada već postoji POW? Odgovor će vas možda začuditi: EXP je potreban „galaksiji"; bez ove funkcije se, jednostavno, ne može realizovati POW; e^x je funkcija koja se lako razvija u red i tako izračunava dok ćemo, za nekoliko trenutaka, videti da se POW(a,b) izračunava kao EXP(b*LN(a)). U međuvremenu vam ostaje da otkucate PRINT EXP(1) i tako saznate sa kolikom tačnošću vaš računar „pamti" konstantu **e.**.

Logaritamske funkcije

PRINT 20*LN (2.367)

Čitaoci ovih redova, baš kao i njihov autor, verovatno pripadaju generaciji koja ne zna šta su to logaritamske tablice i koja koristi kalkulator umesto šibera. Pa dobro, šta su to logaritmi?

Uopšte govoreći, logaritam je funkcija koja ima dva argumenta, **a** i **b**, i koja se piše u obliku log a. Obzirom da naša štamparija ponekad pravi probleme sa indeksima, unekoliko ćemo promeniti ovu konvenciju i pomenuti logaritam pisati kao log (b) a. Ovde je sa **a** označen broj čiji se logaritam traži, a sa **b** takozvana „baza logaritma". Kažemo da je log (b) a=c ako je b^c=a. Logaritmovanje je, dakle, operacija inverzna stepenovanju, baš kao što je deljenje inverzno množenju.

Sve će postati mnogo jasnije kada pogledamo jedan primer. Već smo videli da je 2^3=8, što znači da je log (2) 8=3. Funkcija log može da nam posluži da nađemo eksponent (ovde 3) ako nam je data osnova (2) i rezultat (8). To samo po sebi i nije mnogo korisno, ali treba upoznati još neke fascinantne osobine ove funkcije. Jeste li, na primer, znali da je log (a*b)=log a+log b? Da bismo, dakle, pomnožili dva broja treba da nađemo logaritam svakoga od njih, saberemo te vrednosti i, kako bi se to nekada reklo, „antilogaritmujemo rezultat". Znajući da se logaritmi i „antilogaritmi" lako pronalaze u tablicama i da je lakše sabirati nego množiti, vidimo da logaritmi mogu nečemu i da posluže.

Još bolju primenu su logaritmi nalazili kod stepenovanja. Treba, naime, znati da je log (a^b)=b*log a, pa ćemo biti u stanju da zamenimo stepenovanje realnih brojeva (čak i PI^PI) množenjem koje, ako želimo da idemo do kraja, možemo zameniti sabiranjem i još jednim logaritmovanjem.

Pažljiv čitalac će primetiti da smo u zadnja dva pasusa izneverili konvenciju koju smo uveli samo trenutak ranije: umesto log (b) a pisali smo log a. Šta to može da znači i koliko, najzad, argumenata ima funkcija logaritam?

Oblik koji smo najpre napisali je, strogo posmatrajući, pravilniji: logaritam ima dva argumenta, od kojih je jedan baza logaritma. Pokazalo se, međutim, da se u praksi koriste logaritmi sa bazama 10 (LOG ili dekadni logaritam) i e (LN ili prirodni logaritam). Računar „galaksija" poseduje samo LN kao češće korišćen logaritam.

Šta da radite ako vam nekada zatreba dekadni logaritam ili logaritam za neku drugu bazu? Nema većih problema ako zapamtite formulu log (B) A=LN(A)/LN(B) ili otkucate program poput sledećeg:

```
10 P. „KOJI SE BROJ LOGARITMUJE";
20 INPUT A
30 P. „KOJA JE BAZA";
40 INPUT B
50 C=LN(A)/LN(B)
60 PRINT „LOG (";B;")";A; "="; C
```

Ostalo je da kažemo još par reči o „antilogaritmovanju" i da izložimo jedno ograničenje matematičke prirode. EXP je funkcija inverzna LN, pa bi PRINT EXP(LN(×)), za bilo koje X, trebalo da dâ isto to X u granicama računske greške.

Ovo 'svako X' ne treba shvatiti baš bukvalno: argument funkcije LN mora da bude broj veći od nule, što je posledica definicije logaritma i činjenice da je a^b uvek pozitivan broj.

Završetak asemblera

Iza poslednje asemblerske naredbe, kao što smo videli u prethodnom primeru, mogu da slede definicije i početne vrednosti nekih promenljivih, poruke i tabele. Iza poslednje od njih treba dodati programsku liniju u kojoj se nalazi jedino znak > (veće). Znacima > i < je, dakle, oukviren mašinski deo programa. Kada „galaksija" naiđe na >kontrola se vraća bejzik interpretatoru, dok se oridžin i vrednosti svih labela trajno „zaboravljaju" (ove vrednosti se, istina, ne brišu, tako da bi vešt programer mogao da iskoristi neku od njih).

Iza kraja asemblera može, kao u prethodnom primeru, da se nađe poziv upravo asembliranog programa. Umesto toga, početnici će verovatno radije započinjati osnovni bejzik program koji će docnije i po potrebi pozivati asemblirane potprograme. Kada završite razvoj čitavog programa, sačuvajte izvorni oblik, a zatim, pomoću DEL, obrišite čitav asemblerski deo. Mašinske potprograme pripremite tako da se nalaze ispred bejzika, a zatim snimite samo ono što je neophodno za izvršavanje programa. Na taj način će i oni koji nemaju ROM 2 moći da koriste vaš program. Pored toga, samo će zauzimati manje memorije, pa će se brže učitavati sa kasete.

Postavljanje prekidnih tačaka

U toku razvoja mašinski potprogram neće, jasno, proraditi „iz prve". Moraćete, dakle, da radite na njegovim ispravkama i modifikacijama. U takvim slučajevima, biće vam neophodno da „zavirite" u sadržaj registara i flegova u nekoj fazi izvršavanja programa. Za to vam služi naredba REG.

Postavite REG u neki mašinski program i izvršite ga. Kada mikroprocesor naiđe na REG, prikazaće izveštaj koji se sastoji od sledećih elemenata:

```
AF   BC   DE   HL   1×   SP
AF'  BC'  DE'  HL'  IY   (SP)
```

Vidimo, dakle, sadržaje osnovnih i alternativnih registara, oba indeks registra, vrednost stek pointera i poslednji broj koji je stavljen na stek. Računar čeka da pritisnemo bilo koji taster (osim BRK, DEL i LIST) da bi nastavio izvršavanje našeg mašinskog programa.

Iza REG možemo da napišemo dva argumenta razdvojena zarezom. U tom slučaju, dobijamo i damp segmenta memorije koji nas interesuje. Brojevi iza REG predstavljaju početnu adresu bloka i broj redova koje treba prikazati — baš kao kod naredbi DUMP i LDUMP.

Dejan Ristanović

Ulazne adrese rutina u ROM-u

U sledećoj tabeli su date ulazne adrese važnih potprograma operativnog sistema i ROM-a 2. Razlike u odnosu na tabelu objavljenu u „Računarima u vašoj kući broj 2" se svode na dodavanje ulaznih adresa matematičkih funkcija. Kod svih trigonometrijskih funkcija i kod POW po pozivu potprograma čija je ulazna adresa data DE treba da pokazuje ASCII string u memoriji koji predstavlja argument u zagradi. Za ostale funkcije argument treba da se nađe na aritmetičkom steku.

Šiftovanjem na desno za osam bita dobijamo:
0000 0000 0010 1101=&002D.

Kada bismo ovaj broj šiftovali levo za osam bita, dobili bismo &2D00 što znači da promenljivoj STRANA vrednost možemo da dodelimo i sa:
STRANA EQU RADNI >8<8.

U izrazima može da se pojavi i znak za dolar ($) koji ima sasvim specijalnu funkciju — njime je definisana pseudo promenljiva koja ima vrednost koju će imati PC kada program, u toku izvršavanja, dođe na to mesto. Naredba LD A,($+5) će dovesti u akumulator sadržaj pet bajta udaljene memorijske lokacije. $, dakle, donekle odgovara naredbi PTR bez adresnog dela u bejziku.

Asemblerske naredbe BYTE, WORD i TEXT

U okviru asemblerskog programa ne sme se nalaziti PRINT, IF ili neka druga bejzik konstrukcija. Asembler, međutim, omogućava umetanje nekih naredbi koje Z80 ne bi razumeo. Dve takve smo već upoznali: ORG i OPT. Preostale tri su BYTE, WORD i TEXT (sličnost sa bejzikom je namerna).

Ponekad nam je potrebno da u nekom segmentu mašinskog programa umetnemo neki bajt ili par bajtova koji neće predstavljati kodove neke instrukcije. Potrebno nam je, na primer, da definišemo internu promenljivu SPACE koju ćemo koristiti za privremeno smeštanje sadržaja nekog šesnaestobitnog registra. Na kraju mašinskog programa ćemo, posle poslednjeg RET-a, dodati:
1580 SPACE WORD &2C3A

Time smo definisali dva bajta kojima možemo da se obraćamo sa LD HL,(SPACE), LD (SPACE), DE ili nekom sličnom naredbom. Početna vrednost promenljive SPACE je &2C3A, ali treba obratiti pažnju da će ona ovu vrednost imati samo pri prvom startovanju programa; za docnija startovanja ćemo sami morati da je postavimo.

BYTE, WORD i TEXT se, dakle, koriste za definisanja bajta, reči ili niza bajtova u toku asembliranja. Sledeći jednostavan program će vam pomoći da shvatite upotrebu ovih naredbi.

```
 10
 20 ORG &3000
 30 OPT 3
 40 DFILE EQU &2800
 50 LD A, &C
 60 RST &20 ! SISTEMSKI POZIV ZA BRISANJE EKRANA
 70 LD HL, PORUKA
 80 LD DE, DFILE+231: ! KAO PRINT AT 231
 90 LD BC, KRAJ—PORUKA
100 LDIR
110 RET ! POVRATAK U BEJZIK
120 PORUKA BYTE " " "
130 TEXT „GALAKSIJA"
140 BYTE " " "
150 TEXT " 8 K ROM"
160 KRAJ
170 >
180 A=USR(&3000)
```

Otkucajte i izvršite program pa će se na sredini ekrana pojaviti tekst „GALAKSIJA" 8 K ROM. Primetite da je pojava navodnika u poruci omogućena dvema naredbama BYTE u kojima je korišćena konvencija o navodnicima i to na pomalo čudan, ali za računar sasvim prihvatljiv način.

Trigonometrijske funkcije

```
PRINT SIN (2*PI/3)
PRINT SIND (60)
A=TG (PI/2)
PRINT TGD (45)
A=COS(PI/3)
PRINT COSD(30)
```

Trigonometrija . . . Zvuči zastrašujuće! Bilo bi lako preskočiti ovo poglavlje i preći na mnogo lepši naslov 'proširenja bejzika'. Ipak, obećavamo da poglavlje neće biti teško za čitanje i da ne zahteva neko posebno predznanje. Ako izdržite do kraja, bićete u stanju da, na primer, sastavite program koji na ekranu crta vaš bioritam!

Trigonometrija je prilično stara geometrijska disciplina koja se bavi rešavanjem trouglova. Pod rešavanjem trouglova podrazumevamo računanje svih stranica i uglova jednog trougla kada su neki njegovi elementi poznati. Da bi rougao bio potpuno određen, treba poznavati tri njegova elementa. To ne mogu da budu tri ugla (jer je, znajući da je zbir uglova u trouglu 180 stepeni, treći ugao određen čim su poznata dva) niti dve stranice i ugao naspram manje od njih (jer tada mogu da se konstruišu dva trougla koji zadovoljavaju uglove); bilo koja druga kombinacija je dobra. Pokušajte, na primer, da izračunate uglove trougla čije su stranice 10, 20 i 25 cm. Ne ide, zar ne? Videćete da se problem rešava sasvim jednostavno, uz elementarno poznavanje trigonometrije.

Pokušaćemo, najpre, da „rešimo" pravougli trougao sa prve slike. Jedan od uglova pravouglog trougla je poznat (90 stepeni) pa ga određuju dva elementa. Ukoliko znate dve stranice, lako ćete da odredite treću znajući Pitagorinu teoremu koja, „hakerski" iskazana, glasi C=SQR(a*a+b*b). No, kako da odredite uglove trougla? Ili, kako da odredite dve stranice ako je poznata treća i jedan od uglova?

Potrebne su vam formule:
sin A=a/c cos A=b/c tg A=a/b

Ovde je sa A označen ugao između stranica **b** i **c;** slovo 'alfa' bi možda bolje poslužilo, ali je ovako lakše jer vaša „galaksija", kao i većina računara, nema grčka slova. Bilo kako bilo, sinus ugla je količnik dužina naspramne katete i hipotenuze, kosinus je količnik dužina nalegle katete i hipotenzue, a tangens količnik dužina naspramne i nalegle katete. Setimo se malopređašnjeg problema. Neka je ugao A 20 stepeni a stranica **b** 10 cm. Tada je c=b/cos A i a=b*tg A. Bilo koji džepni kalkulator sa funkcijama će nam reći da je, iz ovih formula, c=10.6418 cm i a=3.6397 cm.

Pre nego što napišemo program koji će rešiti problem, treba da upoznamo novu jedinicu za izražavanje uglova — radijan. Izražavanje uglova u radijanima ima mnoge prednosti u odnosu na izražavanje u stepenima kada se radi o numeričkoj analizi i, posebno, rešavanju transcedentnih jednačina. Za početnika je jednostavnije da uglove izražava u stepenima, jer je na njih navikao. „Galaksija" omogućava obe jedinice: ukoliko je ugao izražen u radijanima, njegov sinus ćemo računati sa X=SIN(A). Ukoliko je A u stepenima, koristićemo

X=SIND(A) (D dolazi od reči „degree", stepen). Na raspolaganju su nam i funkcije CO8, COSD, TG i TGD. Ukoliko nam nekada zatreba da izvršimo konverziju manualno, 180 stepeni je isto što i PI radijana. „Galaksija" može da nam pomogne pri ovoj konverziji — ROM 2 dodaje pseudo promenljivu PI koja ima vrednost ove poznate konstante. Prvih desetak decimala broja PI je 3.141592654, a vi otkucajte PRINT PI da vidite kako ga „galaksija" pamti.

Sledeći program ilustruje neke osnovne trigonometrijske relacije na „galaksiji":

```
10 PRINT „UNESI KATETU B"; : I. B
20 PRINT „UNESI UGAO A"; : I. A
30 C=B/COSD(A)
40 PRINT „HIPOTENUZA JE"; C
50 Z=B*TGD(A)
60 PRINT „KATETA A JE"; Z
70 Y=90-A
80 PRINT „UGAO B JE"; Y
```

Znanje koje smo stekli nam i dalje ne omogućava da rešimo svaki pravougli trougao. Ponekad ćemo, naime, iz formula koje imamo izračunati sin, cos ili tg nekog ugla i poželeti da odredimo sam taj ugao. Za to nam služe inverzne trigonometrijske funkcije arcsin, arccos i arctg (čita se 'arkus sinus', 'arkus kosinus' odnosno 'arkus tangens'). PRINT TG(ARCTG(X)), u granicama računske greške, daje X za bilo koju vrednost promenljive X.

PRINT ARCTG (PI/4)

Poput većine računara, „galaksija" od inverznih trigonometrijskih funkcija ima jedino arkus tangens. Razlog leži u tome što je izuzetno jednostavno naći arkus sinus i arkus kosinus nekog ugla ako znamo njegov arkus tangens. Važi, naime, arcsin x=arctg (x/sqr(1−x*x)), odnosno arccos x=arctg (sqr(1−x*x)/x). Vama ostavljamo da napišete program koji bi ovo potvrdio; za kontrolu će vam poslužiti neki kalkulator ili urođeni osećaj za proporcije u trouglu.

Dva upozorenja vezana za inverzne trigonometrijske funkcije: ARCTG vraća ugao u radijanima. Ako želite rezultat u stepenima, iskoristite PRINT ARCTG(X)*180/Pi. Obzirom na opšte poznatu činjenicu da je bilo koja kateta u trouglu kraća od hipotenuze, sinus i kosinus ugla su uvek manji od jedan (za tangens ovo ograničenje ne važi). Ako pokušate da simulirate arcsin x, gde je x veće od 1, dobićete poruku o grešci jer će računar pokušati da nađe koren negativnog broja (1−x*x je manje od nule ako je x veće od jedan).

Došli smo, najzad, i do rešavanja bilo kog trougla. Za to nam služe sinusna (a/SIN(A)=b/SIN(B)=c/SIN(C)) i kosinusna (a*a=b*b+c*c — 2*b*c*COS (A)) teorema. Iako su relacije jednostavne, predložili bismo vam da zavirite u neki udžbenik srednjoškolske geometrije pre nego što ih primenite; ne možete, na primer, da očekujete da nađete uglove trougla čije su stranice 10, 20 i 100 cm jer takav trougao, jednostavno, ne može da postoji! Postoje mnoga ograničenja poput ovoga; ukoliko ih ne poznajete, vaša „galaksija" će stalno ispisivati WHAT? i HOW? iako je program ispravan.

Funkcija ABS

PRINT ABS (100)
PRINT SQR (ABS(A))

Uspeli smo da grupišemo matematičke funkcije u nekoliko poglavlja, ali je najjednostavnija od njih ostala sasvim usamljena. ABS (x) daje, naime, apsolutnu


```
40 SCF
50 ULAZ LD A, &10
60 ! OSTATAK PROGRAMA
RUN
```

labela ULAZ dobija vrednost &3001 (lokacija &3000 je popunjena kodom instrukcije SCF). Labele koje su dobile vrednost na ovaj način se najčešće koriste kod CALL i JUMP naredbi.

Asembler je opremljen potprogramima za obavljanje računskih operacija koje su potrebne pri radu sa mašinskim jezikom. Sva aritmetika je, po prirodi, celobrojna i nije povezana sa aritmetikom koju koristi bejzik; bejzik promenljive nisu raspoložive u asembleru, kao što labele nisu raspoložive u bejziku.

Izračunavanje izraza se vrši sa leva u desno, pri čemu zagrade nisu dozvoljene. U izrazu mogu ravnopravno da se koriste celobrojne konstante i imena labela. Na raspolaganju su sledeće operacije:

+	celobrojno sabiranje
−	celobrojno oduzimanje
‖	logičko „i" (AND)
<n	šiftovanje na levo za n mesta
>n	šiftovanje na desno za n mesta

Sabiranje i oduzimanje ne zaslužuju poseban komentar. Minus može da se koristi i kao unarni operator promene znaka, mada je jasno da se umesto −2 uvek može iskoristiti &FFFE.

Logičko množenje (AND) se obavlja bit po bit. Da bismo upoznali njegovo dejstvo, proučićemo primer:

```
KRAJ EQU &2C3A &1FA0
```

Prikažimo najpre brojeve &2C3A i &1FA0 binarno. Dobijamo'

```
&2C3A  =0010 1100 0011 1010
&1FA0  =0001 1111 1010 0000
-------------------------------
KRAJ   =0000 1100 0010 0000
       =  0    C    2    0
```

Labela KRAJ je, dakle, dobila vrednost &0C20. Moramo, međutim, da kažemo da se logičko množenje retko koristi za ovako „opšte" operacije — njegova slovna primena je pri takozvanom „maskiranju".

Ponekad će nam, naime, biti potrebno da izdvojimo nekoliko bitova neke vrednosti, a da ostale zanemarimo. Pretpostavimo, na primer, da želimo da labela STRANA dobije vrednost adrese početka strane na kojoj se nalazi labela RADNI (svaka strana memorije ima 256 bajtova). Ukoliko, na primer, RADNI ima vrednost &2DAD, STRANA treba da dobije vrednost &2D00. Potrebno nam je, dakle, da izdvojimo osam signifikantnijih bitova labele RADNI. Za to će nam poslužiti:

```
STRANA EQU RADNI‖&FF00
```

&FF00 će (napišite brojeve binarno i to proverite) zakloniti niži bajt labele RADNI (bilo koji broj logički množen sa nulom daje opet nulu), dok će &FF ostaviti više bajt nepromenjenim.

Šiftovanje ćemo takođe upoznati na primeru:

```
STRANA EQU RADNI >8
```

Neka RADNI ima vrednost &2DAD kao u prethodnom primeru. Prikazan binarno ovaj broj glasi:

&2DAD=0010 1101 1010 1101

Ukoliko nam je, na primer, potrebno da se kod ne upisuje u memoriju i da se listing ne prikazuje na ekranu (prvo isprobavanje programa da bi se jednostavno ispravile sintaksne greške) upotrebićete OPT 0. Ukoliko vam je potrebno da se kod upiše u memoriju uz izdavanje listinga na ekranu (ovu opciju koristite u toku razvoja i testiranja programa), upotrebićete OPT 3 (3=1+2). Ukoliko ste finalizovali neki program vratićete se na OPT 2, kako korisnik ne bi bez potrebe gledao listing koji će ga možda zbunjivati. Za vašu upotrebu ćete, uz OPT 5, ispisati program na ekranu i štampaču.

Ukoliko ne navedete OPT, podrazumevaće se OPT 1. To znači da će program biti asembliran i njegov listing prikazan na ekranu zajedno sa porukama o greškama ali rezultujući kod neće biti upisan u memoriju da ne bi, usled neke greške korisnika, uništio njegov izvorni program. Za stvarnu upotrebu asemblera morate, dakle, da navedete OPT 2 ili OPT 3.

Kako se listing prikazuje? Sa leve strane ekrana vidite adresu instrukcije posmatrano apsolutno (eventualno relociranje se ovde ne prikazuje) u memoriji. Sledi nekoliko bajtova koji predstavljaju kodiranu instrukciju (već smo rekli da neke instrukcije zauzimaju jedan a neke, zajedno sa adresnim delom, čak četiri bajta). Iza toga sledi mnemonik instrukcije i eventualni komentar koji je mogao da pređe i u sledeći red. Labele su posebno istaknute, tako što je ostatak mnemonika uvučen za dva mesta.

„Galaksija" će, prirodno, prijaviti greške u slučajevima kada neka viša akcija nije bila propisna. WHAT? je standardna poruka koja će se pojaviti kada upotrebite nepostojeću instrukciju, HOW? će nastupiti ako, na primer, upotrebite JR tako da pokazuje na instrukciju koja je izvan domena relativnog skoka a SORRY ako se tabela vrednosti labela „sudari" sa bejzik programom. U svakom slučaju biva prikazana linija u kojoj je nastupila greška sa upitnikom na njenom početku. Iskoristite standardnu „galaksijinu" naredbu EDIT ili prekucajte pogrešnu liniju a zatim ponovo startujte program.

Labele i aritmetika

„Galaksijine" labele su reči proizvoljne dužine (sva slova su značajna tj. labela RACUN se razlikuje od labele RACUNAJ) koje počinju slovom a dalje se sastoje od slova i brojeva. Svakoj od njih se dodeljuje ceo broj uz poštovanje „galaksijinih" internih konvencija o kojima smo već dosta govorili. Rezervisane reči asemblera ne smeju da se koriste kao imena labela (ako iskoristite labelu HALT računar neće moći da je razlikuje od odgovarajuće mašinske instrukcije) ali o tome ne treba mnogo da brinete: upotrebite labelu koja vam je potrebna a računar će prijaviti grešku ako mu njeno ime ne odgovara. Primenom / lako ćete locirati sve greške i promeniti nazive.

Vrednost može da se dodeli labeli samo jednom u toku asembliranja i to isključivo na jedan od dva načina: pomoću EQU ili prostim navođenjem na početku neke linije.

Koristeći EQU dodeljujemo labeli vrednost neke konstante ili izraza. Opšti oblik je:

„labela„ EQU „izraz„ (npr. START EQU &2C3A).

Navodeći labelu na početku neke linije dodeljujemo joj vrednost koju će registar PC imati u trenutku kada naiđe na to mesto pri docnijem izvršavanju tog programa. U programu:

```
10 <
20 ORG &3000
30 OPT 3
```


vrednost ili modul argumenta. PRINT ABS (10.25) i PRINT ABS(−10.25) daju isti rezultat: 10.25. Dalji komentar teško da je potreban.

PROŠIRENJE BEJZIKA

Bejzik interpretator koji je ugrađen u ROM 1 je pisan sa težnjom da zauzme što manje memorijskog prostora. Zbog toga su korisnici „galaksije" lišeni određenog broja standradnih naredbi i funkcija. Uložen je, međutim, maksimalan trud da se ostave one naredbe koje će omogućiti veštom programeru da sintetizuje sve bez čega je ostavljen. Dodatkom ROM-a 2 situacija se unekoliko popravlja ali su i njegova 4 kilobajta bila jedva dovoljna za asembler i brojne matematičke funkcije. Ponovo smo bili u situaciji da biramo bejzik dodatke koji će biti od najveće moguće pomoći, ali pre svega onima koji žele da pišu mašinske programe.

Brisanje segmenta i prenumeracija

DEL 100,200

Mašinski programi se, u odnosu na odgovarajuće bejzik rutine, sastoje od velikog broja kratkih naredbi. Obzirom da, kao što ćemo videti, svaka programska linija sme da sadrži samo jednu asemblersku instrukciju, očekujemo programe sa mnogo linija. Ukoliko poželite da obrišete veći segment ovakvog programa, naći ćete se u velikoj nevolji. Lepo je što znate da brojite od 1000 do 10000 u intervalima po 10, ali nije praktično da to znanje potvrđujete previše često! Nikakva programerska „caka", pa čak ni normalna upotreba naredbi BYTE i WORD, ne može da obriše veliki segment linija. To je dovoljan razlog da u ROM-u 2 podržimo naredbu DEL (DELETE=obriši).

DEL ima dva argumenta koji se razdvajaju zarezima. DEL 1000,10000 će, na primer, obrisati sve linije između 1000 i 10000 uključujući i ove dve. Napominjemo da je DEL destruktivna funkcija — ne postoji apsolutno nikakav način da povratite linije obrisane pomoću nje. Pre nego što pritisnete ENTER, treba, dakle, da dobro proverite granice i uverite se da u okviru neke od njih ne postoji višak ili manjak cifre (100 umesto 1000 će vas „zaviti u crno"!). Da bi se bar unekoliko umanjio rizik koji nosi ova naredba, „galaksija" strogo proverava njenu sinteksu: ukoliko ma koja od dve linije pomenute u DEL ne postoji, biva prijavljeno HOW? bez ikakvih loših posledica po program u memoriji.

REN 100

Programiranju u asembleru namenjena je i naredba REN (RENUMBER=prenumeriši). U asemblerskim programima se, naime, koriste samo labele, pa linijski brojevi nisu mnogo bitni. Ovi brojevi, međutim, ponekad mogu da budu i velika smetnja: pišući mašinski program, često ćemo poželeti da umetnemo pedeset instrukcija između dvadesete i dvadeset pete linije!

REN 100 će nam pomoći u takvim situacijama: ova naredba vrši prenumeraciju celog programa tako da linijski brojevi budu 100, 200, 300 . . . Umesto REN 100, možemo da koristimo REN 10 (linije će biti 10, 20, 30 . . .), REN 50 (50, 100, 150 . . .), REN 72 ili nešto slično.

Naredba REN ne sme da se koristi za prenumeraciju bejzik programa. Za ove programe su linijski brojevi veoma bitni: ukoliko je postojala naredba GOTO 125, a linija 125 posle prenumeracije dobije obeležje 500, GOTO 125 će dovesti do greške HOW? ili do potpuno pogrešnog izvršavanja programa. REN će, dakle, pretvoriti bejzik program koji dobro radi u galimatijas, osim ako se odlučite da ručno promenite sve GOTO i CALL i neke TAKE naredbe i tako ih uskladite sa novim brojevima. Obrzirom da „galaksija" nema naredbu AUTO, koja bi olakšala kucanje programa sa „okruglim" linijskim brojevima, ne čini nam se da je trud utrošen na prenumeraciju dužeg programa (čak i kad on treba da bude objavljen u nekom časopisu) uopšte isplativ!

Potrudili smo se da vas na najbolji mogući način zaštitimo od slučajnog kucanja naredbe REN. Pre svega, „galaksija" interpretira R. kao RUN a RE. kao RETURN. čak i ako otkucate REN, računar će prijaviti grešku ukoliko ne sledi propisan broj (normalno bi bilo da se, ako broj nije naveden, podrazumeva 10). Kada, međutim, otkucate REN n i pritisnete ENTER, povratka na staro nema — najbolje je, zato, da program prethodno snimite na kasetu.

Treba pomenuti još jednu „sitnicu": linijski brojevi naredbi mogu da budu najviše 32767; ukoliko upotrebite REN ne uvažavajući ovo ograničenje (REN 250 kod programa koji ima nekih 140 linija) dobićete program na čijem su kraju linije sa negativnim brojevima! U ovakvom slučaju možete da spasete program: jednostavno ponovo otkucajte naredbu REN sa nekim manjim brojem iza nje.

Konvencija o navodnicima

A=„X"

Navodnici se u bejziku koriste kao graničnici alfanumerika. Nema, zato, smisla napisati A=10+„X" jer broj i string ne smeju da se sabiraju. Ukoliko, međutim, otkucate ovakvu naredbu kada je u vašoj „galaksiji" ROM 2, neće se pojaviti nikakvo WHAT?; računar će prikazati uobičajeno READY. Kolika je vrednost promenljive A? Otkucaćemo PRINT A i dobiti 98. Stvari počinju da bivaju jasnije: ASCII kod slova X je 88 a „galaksija" mu je dodala 10.

Prilikom izračunavanja nekog aritmetičkog izraza možete, umesto broja ili funkcije, iskoristiti navodnike između kojih se nalazi neki string. Funkcija će vratiti ASCII kod prvog znaka u tom stringu. Umesto da, na primer, kucate CP 88 u nekom mašinskom programu, napisaćete CP „X". Na taj način ćete najpre sebi olakšati posao, a zatim omogućiti nekom drugom da razume čemu ta naredba služi: vi, zapravo, ne poredite akumulator sa 88, već ASCII kodom slova X_f— zašto da to i ne naglasite?

Sličnom logikom možete da se vodite i pri pisanju bejzik programa: BYTE &29FF„A" će ispisati na samom kraju ekrana slovo A, ne izazivajući pomeranje slike (ovo iz programa ne može da se postigne ni na koji drugi način).

Problem može da nastupi ako želite jednostavno da znate ASCII kod nekog znaka. Ne smete da otkucate PRINT „X", jer ćete na ekranu dobiti obično slovo X. Jedno od rešenja je da otkucate A=„X": PRINT A ali postoje i dva duhovitija:

Otkucajte PRINT 0+„X" i pritisnite ENTER. „Galaksija" će, analizirajući uneti izraz, najpre detektovati nulu kao brojni podatak. Odatle će izvesti zaključak da se ne očekuje alfanumerik i pozvati sistemski potprogram za izračunavanje vrednosti izraza. Ovaj potprogram će, jasno, znati da je 0+„X" isto što i 0+88 i prikazati na ekranu ASCII kod slova X.

Drugi način je još lakši — otkucaćemo PRINT %„X" i dobiti &0058. ASCII kod slova X je, dakle, &58 što je, kada se prevede, isto što i 88 dekadno. Dobra ili loša (zavisi kako posmatrate stvar) strana ovoga metoda je, dakle, što ASCII kod biva izražen heksadecimalno. Potrebno je, ipak, da malo bolje objasnimo znak % koji smo upravo upotrebili: i on je jedna od novina koje donosi ROM 2.

Prikazivanje heksadecimalnih brojeva

PRINT% (&2C3A)

PRINT &3F će, kao što znamo, dati prevod heksadecimalnog broja 3F u decimalan zapis. Kada je god u nekom izrazu moguće napisati decimalan broj, možete da napišete &i neku heksadecimalnu konstantu (svako pravilo ima po neki izuzetak, pa tako i ovo: heksadecimalna konstanta ne sme da se navodi iza NEW i OLD). Obrnuta konverzija, međutim, nije moguća na standardnoj „galaksiji". Dok radite sa mašinskim jezikom, često ćete poželeti da je primenite, jer većina tabela operiše sa heksadecimalnim brojevima. Zato je znak za procenat (%) rezervisan za prevođenje decimalnih brojeva u heksadecimalne. Jasno je da ovaj znak može da ima smisla jedino kod PRINT naredbe.

Područje video memorije i sistemskih promenljivih je dostupno ali opasno: video memorija se stalno menja ako zahtevamo listing programa, područje numeričkih varijabli i bafer za tastaturu su dostupni bez opasnosti, dok je ostatak manje-više zabranjen. Programi se obično upisuju od &2C3A ali ne treba zaboraviti da u tom slučaju bejzik treba da bude pomeren sa NEW 100, NEW 1000 ili nečim sličnim; u protivnom će rastući mašinski program „pojesti" svoj izvor sa nepredvidljivim ali uvek vrlo neprijatnim rezultatima.

Smeštanje mašinskog programa na sam kraj memorije je sasvim moguće, ali uz razumevanje jednog ograničenja. „Galaksiji" je potreban prostor za smeštanje imena labela i njihovih adresa. Za ovo se koristi prostor ispod RAMTOP-a. Ukoliko ovde smeštate mašinac nastaće totalna konfuzija. Osim toga, asembliranje će svakako uništiti određen broj elemenata alfanumeričke matrice X$(I) i/ili elemenata numeričkog niza A(I).

Problem se rešava na jedan od dva načina. Pre asembliranja možete da promenite RAMTOP (sistemska promenljiva na adresi &2A6A) i tako zaštite važne podatke ili zonu na kraju memorije u koju će mašinac biti smešten.

Drugo rešenje je relocirano upisivanje programa u memoriju. Program se asemblira tako da mu je oridžin, na primer, &3FA0 a upisuje &500 bajtova niže. Po završetku asembliranja ovaj program premeštate na pravo mesto primenom FOR-NEXT petlje ili snimate na kasetu, a zatim relocirano upisujete sa OLD 1280. Pri ovakvoj egzibiciji neophodno je dobro paziti na to da ORG pokazuje adresu na kojoj će se program izvršavati (u našem slučaju &3FA0), dok će se relociranje postići primenom komande OPT.

Komanda OPT

Kao i ORG, OPT je opisna komanda koja ne rezultira smeštanjem nekog bajta u memoriju; ona jedino govori računaru koju od opcija asembliranja da izabere.

OPT ima dva argumenta razdvojena zarezom: OPT n,r. Prvi argument je obavezan a drugi neobavezan: moguće je, dakle, upotrebiti OPT n. Suprotno uobičajenom redu stvari, objasnićemo najpre drugi argument jer je neposredno povezan sa sadržajem prethodnog poglavlja.

Slovo r simbolizuje broj bajtova za koji treba relocirati mašinski program pri upisu u memoriju. Sekvencu:

```
10<
20 ORG &2C3A
30 OPT 3,&1000
40! OSTATAK PROGRAMA
```

koristimo da asembliramo program koji će pri izvršavanju biti smešten od &2C3A, ali će u toku asemblilaranja biti upisivan od &3C3A (&2C3A+&1000). Argument r može da bude i negativan (npr. OPT 3,—&100 bi upisivalo program 256 bajta pre ORG-a) uz uvažavanje „galaksijine" predstave brojeva. U granicama o kojima ćemo govoriti u sledećem poglavlju r može da bude i izraz.

Argument n je ceo broj između 0 i 7. Odredićete sa konsultujući sledeću tabelu:

n	Funkcija
1	asemblerski listing se prikazuje na ekranu
2	rezultujući mašinski program se upisuje u memoriju
4	asemblerski listing se ispisuje na štampaču

nekih „specijalnih efekata", kao što je postavljanje prekidnih tačaka u programima.

„Galaksijin" asembler je organizovan tako da se mašinski programi pišu kao deo bejzika. Time je, najpre, ušteđen prostor jer nije bilo neophodno sastavljati novi editor, a zatim olakšano pisanje bejzik programa sa mašinskim potprogramima, što će verovatno najviše interesovati početnike. Da bi se prevazišlo ograničenje nastalo zbog toga što, u ovakvoj organizaciji, nije moguće pisati program koji će se smeštati u svaki segment memorije (npr. programa koji će se upisivati od linka za video), omogućeno je relocirano asembliranje, dopunjeno programom za relocirano snimanje koji smo dali u „Računarima 2".

Mašinske programe pišete, dakle, poput bejzik programa: svaka linija počinje brojem, sledi (neobavezna) labela iza koje mora da se nađe bar jedan blanko simbol. Zatim pišemo standardnu mnemoničku skraćenicu instrukcije, a iza nje, ukoliko je potrebno, adresni deo. Adresni deo je od instrukcije odvojen bar jednim blanko simbolom. Iza instrukcije sledi (neobavezni) komentar koji počinje uzvičnikom. Dozvoljeno je postojanje linija u kojima se nalazi samo labela, ili samo komentar, ali ne i smeštanje većeg broja instrukcija u jednu liniju.

Evo nekoliko primera korektno napisanih linija:

```
100 SCF
100 LD A, (HL)
100 ! POČETAK PROGRAMA
100 CIKLUS PUSH BC
100 PROVERA
100 ! PROVERA
100 PROVERA CP „A" ! DA LI JE A?
```

A zatim i nekoliko nepravilnih linija:

```
SCF                     nedostaje broj
100 LDA,$10             nema blanka između LD i A
100 PRVA PROMENA        blanko u labeli ili komentar bez!
100 SCF:ADC (HL)        dve instrukcije u liniji
100 EX BC, DE           nepostojeća instrukcija
100 1CIKLUS             ime labele ne sme da počinje cifrom
```

Početak asemblera

Asemblerski programi se, kako rekosmo, uklapaju u bejzik. Da bi rekli računaru kada započinje mašinski potprogram, tj. kada treba da aktivira ROM 2, iskoristićemo programsku liniju na čijem se početku nalazi znak (manje). U tu liniju nećemo više ništa upisivati, tako da se u docnijem listingu lako uoči mašinski potprogram. Od momenta kada „otvorimo asembler" više ne radimo na bejziku — naredbe PRINT, STOP, IF i slične će izazvati grešku ili biti interpretirane kao labele. Na raspolaganju su nam jedino komande koje će biti opisane u daljem tekstu.

Prva naredba mašinskog segmenta treba da bude ORG (origin, početak). Iza ove naredbe stavljamo adresu (dekadnu ili, ako joj prethodi &, heksadecimalnu) na koju treba da bude smešten mašinski program. ORG &3000 će, na primer, smeštati mašinski program tako da počinje od &3000, pa ćemo docnije moći da ga pozovemo sa A=USR(&3000). Treba, međutim, da pazimo gde smeštamo mašinski program!

Smeštanje na adrese 0—&1FFF (prostor koji zauzimaju ROM 1 i ROM 2) ne može da donese nikakve probleme ali ni koristi: „galaksija" pokušava da upiše nešto u ROM i u tome, na svu sreću, ne uspeva. Smeštanje programa u prostor latcha i tastature je korisno jedino ako želimo da vidimo naš računar u haosu.

PRINT % izraz će, dakle, izračunati (izraz) i pretvoriti ga u četvorocifreni heksadecimalni broj koji će biti prikazan na ekranu uz vodeće & kako bi se izbegla zabuna. Neke od vodećih cifara mogu da budu nule, i to ponajviše zbog „galaksijine" konvencije o celim brojevima.

Sa korisničke tačke gledišta, „galaksija" ne operiše sa celim brojevima: sve promenljive čuvaju brojeve u takozvanom pokretnom zarezu. Za interne potrebe „galaksija", međutim, koristi cele brojeve prikazane binarno u potpunom komplementu (ukoliko vam reči „potpuni komplement" i „pokretni zarez" ništa ne znače, zanemarite ostatak ovoga pasusa). Brojevi manji od &8000 su pozitivni (bit 7 više bajta veće težine resetovan), a konstante veće do &8000 su negativne. Zato će PRINT MEM, ukoliko imate memorijsko proširenje od 48 Kb, dati utisak ne samo da nemate memorije, već i da je nekome dugujete! Ovaj problem rešavate tako što uz ROM 2 kucate PRINT %MEM i interpretirate dobijenu heksadecimalnu vrednost kao broj slobodnih bajtova ili, ako ne želite da imate posla sa heksadecimalnim brojevima, PRINT MEM+65536 (odakle li je došao broj 65536? To je 2 16 ili 64 Kb). Na sličan način možete da koristite i PRINT %WORD(&xxxx) i tako izbegnete mnoge neprijatne situacije.

Prikazivanje sadržaja memorije

DUMP &1000,212

Često je potrebno prikazati sadržaj niza sukcesivnih memorijskih ćelija. To može da se učini uz pomoć FOR-NEXT petlje, ali je takvo rešenje „sirotinjsko": sporo je i zahteva mnogo kucanja. Zato je u ROM-u 2 našla mesto naredba DUMP koja ima dva argumenta odvojena zarezom.

DUMP x,n prikazuje sadržaje memorijskih ćelija počevši od one čija je adresa x. Prikazuje se sadržaj osam ćelija u svakom redu, pri čemu se na početku svakog reda nalazi adresa njegove prve ćelije. Biće prikazano n redova (n može da bude najviše 255), tj. dampovano 8*n bajtova počevši od onoga čija je adresa x. Sve konstante se prikazuju heksadecimalno, dok x i n mogu da budu dekadni ili (ako nekom od njih prethodi &) heksadecimalni brojevi, pa čak i izrazi.

Otkucajte, na primer, DUMP &1000,200 i na ekranu će biti prikazano prvih 1600 bajtova ROM-a 2. Cifra će, narevno, biti previše da biste ih pratili (sa razumevanjem tih cifara ćete još malo pričekati — treba da naučite mašinski jezik) pa ćete morati da pritisnete DEL (privremeni prekid prikazivanja) ili BRK (trajni prekid i poruka READY).

Dampovanje memorije se najčešće koristi za traženje nekog dela mašinskog programa. Traženje neke naredbe u bejzik programu je daleko lakše ako pročitate sledeće poglavlje.

Traženje stringa u bejziku

/ „GALAKSIJA"

„Galaksija" čuva bejzik programe u memoriji na jednostavan i prirodan način — bajt po bajt. Zato nije bilo ni malo teško realizovati funkciju koja će vam pomagati pri traženju nekog teksta u bejzik programu. Ovakva funkcija postaje apsolutna nužnost kada se pređe na asembler sa mnoštvom labela.

Želeli smo da se nova funkcija što lakše koristi, pa smo za nju morali da iskoristimo neki nešifrovani specijalni znak. Jedini slobodan je kosa crta (/). Otkucajte, dakle, /PRINT i „galaksija" će izlistati sve programske linije u kojima se nalazi reč PRINT (ne i one u kojima je P., PR. ili nešto slično); ako ih ima previše, možete da koristite DEL da usporite prikazivanje.

Pomoću / možete da tražite bilo koju programsku naredbu, ime promenljive, konstantu, komentar ili, jednostavno, bilo koji deo bejzik programa. Nije moguće tražiti jedino programske linije, ali takvo traženje i nema smisla; ako znate broj neke linije, možete da iskoristite LIST da je vidite.

PODRŠKA PORTA I ŠTAMPAČA

ROM 2 je softversko proširenje „galaksije", što znači da je bilo hardverskih i da će ih biti još više. Takva proširenja se priključuju na port za ekspanziju koji je smešten sa zadnje strane računara. Da bi takva proširenja bila na najbolji mogući način kontrolisana, ROM 2 je opremljen naredbama INP i OUT. Pošto je štampač najpotrebnija periferijska jedinica koja pretvara kompjuter u nešto korisno, podržali smo rad sa njim u ROM-u. Ukoliko u ovom trenutku nemate nikakav uređaj koji biste priključili na vašu „galaksiju", ne smatrajte da je prostor u ROM-u 2 koji opisujemo bačen — vaš računar je donedavno imao i jedno prazno podnožje, pa ste ga popunili ovim ROM-om; na sasvim sličan način ćete uskoro „popuniti" i port!

Naredbe INP i OUT

```
X=INP (10)
OUT 10,&FF
```

Mikroprocesor Z80A može da opšti sa periferijskim uređajima na dva načina: preko memorijske i input/output (ulazno/izlane) mape. Opštenje preko memorijske mape se zasniva na tome da memorijska lokacija &FFFF, na primer, deluje na neki priključeni relej. Tada će BYTE &FFFF,1 uključiti a BYTE &FFFF,0 isključiti uređaj koji je kontrolisan relejom.

Kontrola uređaja uz pomoć memorijske mape nije uvek sjajno rešenje: šta ako imamo pun adresni prostor popunjen ROM-om i RAM-om? Zato su konstruktori mikroprocesora predvideli još jednu, I/O mapu, koja ima 256 ćelija koje nazivamo portovima. Neki periferijski uređaj može, na primer, da bude konstruisan tako da ga upis broja &FF u port nula uključuje, a broja &00 u isti port isključuje. Komunikacija može da bude i dvosmerna: priključeni uređaj može da saopštava neke informacije koje će kompjuter „čitati" iz nekog drugog (ili tog istog) porta. EPROM programator, na primer, može da prima od računara sadržaj koji treba programirati ili da izveštava kompjuter o njegovom sadržaju.

Ni jednom od I/O portova ne možete da pridete naredbom BYTE ili WORD. Da biste iz bejzika pročitali sadržaj nekog porta, koristićete funkciju INP a da biste upisali nešto u port — naredbu OUT.

Funkcija INP ima jedan argument koji se obavezno nalazi u zagradi — broj porta koji je između 0 i 255. PRINT INP(10) će, na primer, prikazati sadržaj desetog porta.

Naredba OUT ima dva argumenta odvojena zarezom. Prvi od njih predstavlja broj porta (0—255), a drugi sadržaj koji treba upisati u port (obzirom da svaki port predstavlja jednu memorijsku ćeliju, i ovaj sadržaj mora da bude između 0 i 255 odnosno, što je isto, 0 i &FF). OUt 100,&F0 upisuje konstantu &F0 u port čiji je broj 100. Umesto ovoga oblika, mogli smo da koristimo i OUT &64,&F0 ili OUT 100,240.

Port broj 255 namenjen je radu sa štampačem i treba izbegavati njegove druge primene. Kada je bit 7 ulaza sa porta &FF resetovan, štampač je spreman za prijem sledećeg znaka, a ako je setovan, „galaksija" treba da sačeka.

Podrška štampača

LPRINT „GALAKSIJA"

Bilo koji standardni matični štampač može da se poveže sa računarom „galaksije" može preko Centronics interfejsa, koji se dokupljuje ili gradi kao posebna opcija, da se poveže sa bilo kojim standardnim matričnim štampačem. Od tog trenutka možemo da štampamo programe, rezultate njihovog izvršavanja, segmente memorije i asemblirane rutine. Sve ovo nam omogućavaju naredbe LPRINT, LLIST i LDUMP koje dodaje ROM 2.

LPRINT je, u suštini, isto što i PRINT, s tim što se sadržaj liste prikazuje na štampaču umesto na ekranu. Iza ove naredbe sledi lista koja liči na onu iza „običnog" PRINT-a, osim u jednoj sitnici: iza LPRINT sme da se nalazi samo jedan element. To može da bude promenljiva, izraz, alfanumerik ili CHR$, ali nije dozvoljeno navoditi nekoliko podataka koji bi bili razdvojeni zarezom ili tačkom i zarezom. Nije, takođe, dozvoljeno korišćenje naredbe LPRINT AT. Ukoliko vam je ona neophodna, ispišite niz blankova ispred teksta koji treba štampati.

Većina štampača prihvata različite kontrolne kodove koji određuju neke njihove funkcije: tip slova, broj redova na strani, širina papira . . . Ovi kodovi se nalaze u uputstvu za korišćenje vašeg štampača i „galaksija" će biti srećna da ih pošalje preko LPRINT CHR$(x). Treba obratiti pažnju na manji bas „galaksijinog" operativnog sistema koji onemogućava da se pošalje CHR$(0). Ovo je bag uobičajen za Microsoftov bejzik, pa svi štampači dozvoljavaju korišćenje nekog drugog koda umesto nule sa istim efektom.

Da bi se na štampač preneo sadržaj ekrana na kome je nešto crtano primenom naredbe DOT, potrebno je posedovati printer opremljen takozvanom „blok grafikom". Kod ovakvog printera kodovi 128—192 daju blokove koji odgovaraju organizaciji „galaksijine" video memorije. Preciznija obaveštenja o ovoj organizaciji su već objavljena i biće ponovljena u okviru uputstva za upotrebu oficijelnog štampača.

LLIST
LLIST 1000

Naredba LLIST omogućava prenošenje čitavog programa ili jednog njegovog dela na papir. Upotreba je jednostavna: otkucajte LLIST i pritisnite ENTER. Listanje će trajati sve dok je ENTER pritisnut; njegovo otpuštanje izaziva privremeni, a pritisak na BRK trajan prekid rada štampača. Iza LLIST može da sledi naredbe od koje listanje treba da počne.

LDUMP 0,10

Naredba LDUMP je potpuno identična već opisanoj DUMP, s tim što se sadržaj šalje na printer i na ekran. LDUMP 0,512 će, na primer, ispisati heksadecimalni sadržaj ROM-a 1.

U podršku štampača spada, strogo gledajući, i jedna od opcija asemblera (OPT 4). Ova mogućnost će biti objašnjena nešto docnije.

Ukoliko ste dovoljno nestrpljivi da isprobate sve pomenute naredbe bez štampača, „galaksija" će se blokirati, ali će je BRK ili RESET vratiti u normalno stanje.

Tretiranje slova Č, Ć, Ž i Š

Set znakova računara „galaksija" je, kao što znamo, doupnjen našim latiničnim slovima. Većina raspoloživih štampača, međutim, ima malo razumevanja za naše probleme ovoga tipa — ASCII kodovi 91—94 koji, u „galaksijinom" setu, odgovaraju našim slovima, izazivaju štampanje specijalnih simbola, najčešće strelica. Listing programa koji sadrži naša slova bi, dakle, izgledao u najmanju ruku smešno!

Da bi se izbegao ovaj problem, „galaksija" umesto slova Č i Ć štampa C, umesto Ž štampa Z dok S zamenjuje Š. Ova konvencija može da bude neprijatno ukoliko nabavite štampač sa našim setom karaktera ili promenite EPROM u štampaču. Ukoliko želite da isključite pomenutu pogodnost, otkucajte BYTE &2AAB,1 a ponovo da je uključite — BYTE 2AAB,0.

ASEMBLER

U ROM 2 je ugrađen originalni asembler napisan uz poštovanje svih Zilogovih konvencija i prilagođen specifičnostima „galaksijinog" operativnog sistema. On omogućava komforno pisanje i ispravljanje mašinskih programa uz podršku

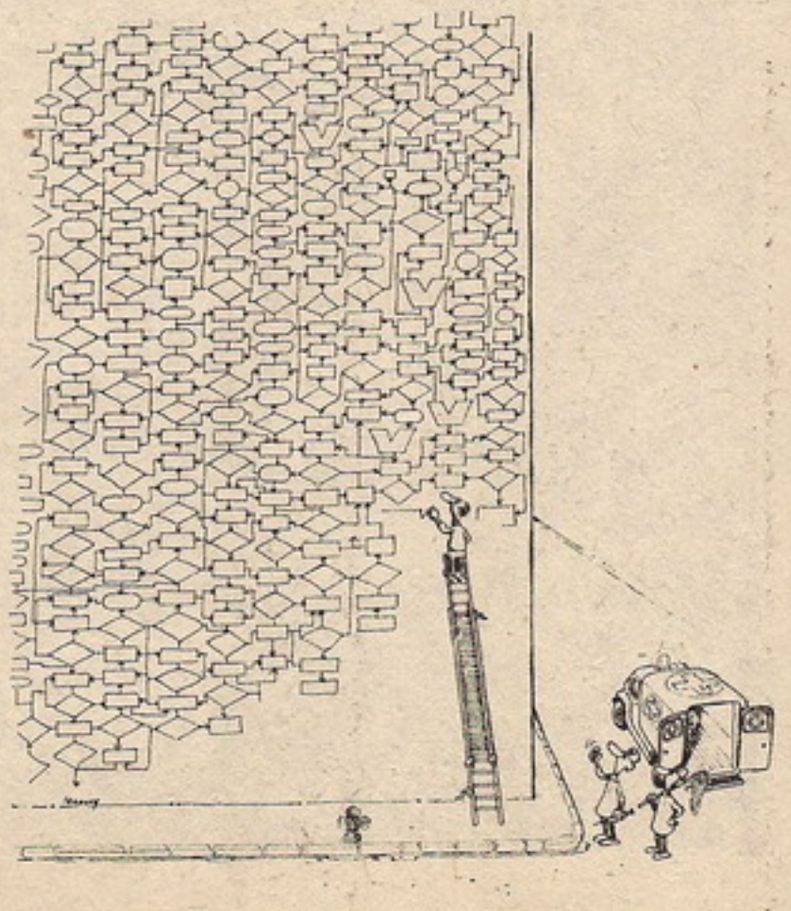
Nastavak sa strane 35

```
 10 REM
 20 REM
 30 REM      Epson printer driver  0.
 40 REM
 50 REM             24.11.1984.
 60 REM
 70 REM          Dejan  Ristanovic
 80 REM
 90 REM
100 UNDERL=129
110 OSBYTE=&FFF4
120 OSWRCH=&FFEE
130 FOR I=4 TO 7 STEP 3
140 P%=&400
150 O%=&6000
160 [ OPT I
170 JMP WORK
180 JMP SWON
190 JMP SWOFF
200 RTS
210 .TEMPA EQUB 0
220 .FLAGS EQUB 0
230 RTS
240
250 .SWON           \ ukljucivanje
260 LDA #3
270 LDX #10
280 JSR OSBYTE
290 LDX #NOSKIP MOD 256
300 BNE PRNT        \ uvek JMP
310
320 .SWOFF          \ iskljucivanje
330 LDX #RESET MOD 256
340 JSR PRNT
350 LDA #3
360 LDX #0
370 JMP OSBYTE
380
390 .WORK           \ obrada slova
400 PHP
410 STA TEMPA
420 TXA
430 PHA
440 TYA
450 PHA
460
470 LDA TEMPA
480 CMP #UNDERL     \ da li je
490                 \ podvlacenje?
500 BNE NUNDERL
510
520 LDX #SUNDERL MOD 256
530 LDA FLAGS
540 EOR #&40        \ da li je
550                 \ podvlacenje vec
560                 \ bilo aktivno?
570 STA FLAGS
580 BNE SEND
590 LDX #RUNDERL MOD 256
600
610 .SEND
620 JSR PRNT
630 BEQ FIN         \ uvek JMP
640
650 .NUNDERL
660 LDX #TABLE MOD 256
670
680 .CYCLE
690 LDA TEMPA
700 CMP &400,X      \ prepoznavanje
710                 \ spec. slova.
720 BNE CONT
730
740 INX             \ slovo prepoznato
750 TXA
760 PHA
770 LDX #CHAR MOD 256
780 JSR PRNT
790 PLA
800 TAX
810 JSR PRNT
820 LDX #ZERO MOD 256
830 BNE SEND        \ uvek JMP
840
850 .CONT
860 TXA
870 SEC
880 ADC #10
890 TAX
900 LDA &400,X
910 BNE CYCLE       \ kraj tabele?
920 LDA TEMPA
930 JSR OSWRCH      \ obicno slovo.
940
950 .FIN
960 PLA
970 TAY
980 PLA
990 TAX
1000 LDA TEMPA
1010 PLP
1020 .RET
1030 RTS
1040
1050 .PRNT          \ stampa seriju
1060                \ kodova na koju
1070                \ pokazuje reg. Y.
1080 LDA &400,X
1090 CMP #&FF
1100 BEQ RET
1110 JSR OSWRCH
1120 INX
1130 BNE PRNT       \ uvek JMP
1140
1150 .RESET         \ inicij. Epsona.
1160 EQUB 27
1170 EQUB ASC "@"
1180
1190 .ZERO
1200 EQUB 0
1210 EQUB 0
1220 EQUB 0
1230 EQUB 255
1240
1250 .CHAR          \ bit image mode.
1260 EQUB 32
1270 EQUB 8
1280 EQUB 27
1290 EQUB 89
1300 EQUB 12
1310 EQUB 0
1320 EQUB 255
1330
1340 .SUNDERL       \ set underline
1350 EQUB 27
1360 EQUB ASC "-"
1370 EQUB 49
1380 EQUB 255
1390
1400 .RUNDERL       \ reset underline
1410 EQUB 27
1420 EQUB ASC "-"
1430 EQUB 48
1440 EQUB 255
1450
1460 .NOSKIP        \ iskljucuje pres-
1470                \ kakanje kraja
1480                \ strane.
1490 EQUB 27
1500 EQUB ASC "O"
1510 EQUB 255
1520
1530 EQUS "(C)84 DR"
1540 .TABLE
1550 ]
1560 NEXT I
1570
1580 FOR I=0 TO 87
1590 READ A:O%?I=A
1600 NEXT I
1610
1620 *SAVE Epson 6000 +100 400 400
1630
1640 END
1650
1660 REM   Za svako slovo se salje
1670 REM   najpre sekvenca CHAR zatim
1680 REM   sekvenca iz DATA liste ko-
1690 REM   ja pocinje njegovim kodom
1700 REM   i, na kraju, sekvenca ZERO.
1710
1720 DATA 91,28,34,128,34,64,34,128,34,0,255
1730 DATA 95,28,34,0,34,64,34,128,34,0,255
1740 DATA 93,34,132,34,72,34,144,34,0,0,255
1750 DATA 92,16,42,128,42,64,42,128,42,4,255
1760 DATA 124,36,18,64,146,64,18,64,146,76,255
1770 DATA 123,60,66,128,66,0,66,128,66,36,255
1780 DATA 96,60,66,0,66,0,66,128,66,36,255
1790 DATA 125,0,66,132,74,16,98,128,66,0,255
>
```

Što se moda 6 i prelaska na mala slova tiče, nema mnogo problema: poziv *FX 202 omogućava proizvoljno stanje 'LOCK' tastera. Kako, međutim, da „otkucamo" NEW i PRINTER XXX, kada smo već startovali View? Rešenje je na prvi pogled čudno, ali i vrlo efikasno: smeštamo tekst NEW i PRINTER XXX (uz obavezne bajtove &D na kraju svake naredbe) u bafer za tastaturu koristeći, *FX 138. Kada „View" preuzme kontrolu, „pomisliće" da smo unapred otkucali ove naredbe i izvršiti ih. Program pisan prema ovim zahtevima dat je na prvoj slici uz obilje komentara.

Sledeći posao je pisanje drajvera za štampač. Ovde treba, najpre, dobro proučiti poslednje poglavlje knjige „Into View" koja opisuje veoma precizni format koji ovaj drajver mora da sledi. Iz ovog poglavlja možemo da zaključimo da drajver mora da ima najviše 256 bajtova mašinskog programa, koji počinje od &400. To znači da moramo da koristimo mogućnosti relociranog asembliranja, koja je objašnjena u ovom izdanju. Funkcija samog drajvera je vrlo jednostvna: „View" će mu se obraćati kada poželi da uključi ili isključi štampač (tada se izvršava VDU 2 ili VDU 3) i kada šalje kod karaktera koji treba da bude štampan. Drajver prepoznaje specijalne definisane znake i, umesto njih, šalje štampaču seriju kodova koji proizvode slova kao matrice tačaka. Ovi kodovi variraju od štampača do štampača, pa će biti neophodno da konsultujete vaše uputstvo za upotrebu i pokažete svoje dizajnerske sposobnosti u crtanju slova. Obratite pažnju na to da drajver dat na slici 2 odgovara štampačima Seikosha GP100 i GP250, kod kojih svakom slovu prethodi &08 za prelazak u grafički mod, a zaključuje ga &80 (prazna linija koja odvaja slova) i &0F (prelazak u tekst mod). Bilo je, takođe, potrebno pomalo žonglirati sa kodom &0D koji treba da izazove prelazak na novi red.

Dejan Ristanović



KOMPJUTORAŠI
što ste tražili to ste dobili!!
Redakcija magazina

SAM

SVOJ MAJSTOR

pripremila je ZA VAS
»DATASET« PROGRAM

SAM DATASET PROGRAM ŠTEDI VAŠE VRIJEME

▼ DATASSETTE kompjutorske kazete C-12 i C-20 u prednosti su pred kazetama standardne dužine. ZAŠTO? Ako radite s kompjutorom, odgovor znate i sami! (Cijena: **C-12 230 d, C-20 260 d**)

▼ DATASET JOTTER smješta svaki pixel na pravo mjesto i dodjeljuje mu točnu koordinatu. Koliko je to važno, shvatit ćete ako se sjetite da na ekranu postoji 49152 pixela. (Cijena: 480 d)

▼ DATASET SPRITE pomaže vam pri kreiranju simbola ili karaktera po vašem izboru. (Cijena: 240 d)

SAM DATASET PROGRAM
ZAISTA ŠTEDI VAŠE VRIJEME!

Kako nabaviti sve što nudi SAMOV DATASET PROGRAM?

1. Pošaljite poštanskom uputnicom za novac odgovarajući iznos na adresu: VJESNIK, Šalter sitne prodaje, 41000 Zagreb, Avenija bratstva i jedinstva 4
2. Na poleđini poštanske uputnice u rubriku »Saopćenje primaocu« svakako naznačite naziv i količinu proizvoda kojeg naručujete;
3. Naručena pošiljka stići će vam na kućnu adresu. Troškovi otpreme i poštarine uključeni su u cijenu proizvoda.

KORISNIĆI

PC ZX 81, SPECTRUM, COMMODORE 64

I OSTALIH EXTERNIH JEDINICA

******** **NOVO** ********

OD 1. NOVEMBRA 1984. GODINE

zagrebcenter — *Progis — Computer Studio*
41000 ZAGREB, Gorjanska 6,
tel. (041) 560-612

Radno vrijeme od 07,00 — 19,00 sati
Subotom od 08,00 — 12,00 sati

*****

Novo otvoreni COMPUTER STUDIO korisnicima Personal i mini computerskih sistema, na jednom mjestu nudi kompletnu uslugu:

— *HARDWARE*
  — *popravak*
  — *servis*
  — *proširenje kapaciteta memorije*
  — *dogradnja za aplikaciju na periferne jedinice*

— *SOFTWARE*
  — *veliki izbor gotovih programa HOBBY programi*
  — *video igre*
  — *logičke igre itd.*

— *PROGRAMI ZA POSLOVNE POTREBE*
  — *knjigovodstveni programi*
  — *statistički programi*

— *KORISNIČKI PROGRAMI PO POSEBNOJ NAMJENI I NARUDŽBI*

*****

Computer Studio nudi zainteresiranim samostalan rad na našim instaliranim sistemima.

Odabrani programi mogu se odmah isprobati, snimiti na diskete ili kazete i odmah preuzeti.

Očekujemo Vaš telefonski poziv ili posjet.

Biblioteka programa Komodor 64

NOVE NAREDBE NA „KOMODORU“

Dobre zvučne i grafičke mogućnosti računara CMB-64 nisu raspoložive iz njegovog standardnog, pa njegovi vlasnici već posle prvih nepisanih programa požele neku moćniju verziju ovog jezika. Za početak nudimo jedno proširenje koje, doduše, znatno manje pruža od „Sajmons bejzika“, ali ga zato može imati svako ko je dovoljno strpljiv da unese ponuđene programe.

U DATA linijama programa 2 zapisan je niz rutina na mašinskom jeziku koji omogućava jednostavno korišćenje zvučnih mogućnosti, rad sa sprajtovima i ekranom i kontrolu džojstika. Rutinama se pristupa korišćenjem SYS naredbe iza koje slede naziv rutine i njeni parametri. Za ulazak u program treba prvo otkucati program 1 i smestiti ga na početak kasete. Zatim se naoružajte strpljenjem i krenite u višečasovnu borbu s programom 2. Preporučujemo da još u toku unošenja više puta izvršite memorisanje na pomoćnu traku i tako izbegnete da više od dve stotine redova izgubite svojom greškom ili zbog nestanka struje na primer. Kada uz sve mere opreza unesete program, startujte ga i ispravite sve greške koje vam budu prijavljene. Uz pomoć četvorocifrenog broja (sem u liniji 1260) na kraju svake DATA linije koji predstavlja kontrolnu sumu, program proverava da li ste korektno uneli sve podatke. Ako se kontrolna suma ne slaže, program se prekida i dobijate broj linije u kojoj ste napravili grešku. Ako ste slučajno umesto zareza u nekoj liniji otkazali tačku između podataka dobićete grešku u liniji 4040. Kada svi podaci budu korektni dobićete poruku da pozicionirate traku. Ona treba da bude postavljena neposredno iza programa 1. Tada memorišite ove mašinske rutine. Za korišćenje novih naredbi premotajte traku na početak i pritisnite istovremeno SHIFT i RUN/STOP. Od tog trenutka nadalje računar će brinuti sam o sebi.

Pre no što damo sintaksu i objasnimo dejstvo instrukcija ponuđenih ovim programom, zadržaćemo se malo na samom bejzik programu napisanom jedino korišćenjem standardnih naredbi. Poznato nam je da su mnoge srednje škole koje obrazuju programere kupile za svoje učenike računare CMB-64 i na taj način bar malo ublažile činjenicu da učimo decu programiranju bez računara. Upravo zbog učenika koji standardan bejzik uče kroz svoje školske programe, analiziraćemo kako radi ovaj program i na taj način ih, nadamo se, podstaći da i sami pokušaju da napišu nešto slično.

Program praktično počinje od linije 5000 porukom da treba da sačekate punjenje podataka. Prvo se preuzimaju podaci iz linija 110—290 o nazivima i adresama uvedenih naredbi. Nazivi se smeštaju od registra 49152, a adrese rutina koje realizuju

Sintaksa i značenje novih naredbi

Redni broj	FORMAT	PARAMETRI	NAMENA
1.	SYS 49664, „VOL“ v	v (0—15) : jačina zvuka	Postavlja jačinu zvuka.
2.	SYS 49664, „ENV“ v, a, d, s, r	v (1—3) : glas (kanal) na koji se odnose karakteristike. a (0—15) : dužina faze pojačavanja d (0—15) : dužina faze stišavanja s (0—15) : dužina faze održavanja r (0—15) : dužina faze otpuštanja	Definiše zvučnu sliku tonova koje treba odsvirati.
3.	SYS 49664, „SIL“	bez parametara	Briše SID čip i sve isključuje. Ovu rutinu treba koristiti pre i posle korišćenja zvuka.
4.	SYS 49664, „PUL“ v, w	v (1—3) : glas (kanal) w (0—4095):	Postavlja trajanje impulsnog talasnog oblika.
5.	SYS 49664, „WAV“ v, w	v (1—3) : glas (kanal) w (0—3) : talasni oblik	Postavlja talasni oblik za određeni glas pri čemu w=0 odgovara trouglastim, w=1 testerastim, w=2 četvrtastim i w=3 nepravilnim (šum) talasima.
6.	SYS 49664, „VOI“ v, p	v (1—3) : glas (kanal) p (0—96) : visina glasa — Ø isključuje	Uključuje i isključuje glas i postavlja visinu.
7.	SYS 49664, „SOU“ v, p	isto kao u prethodnoj naredbi	Uključuje glas na unapred određeno trajanje, a zatim ga isključuje.
8.	SYS 49664, „POI“ s, l	s (0—7) : broj sprajta l (0—16383) : lokacija	Postavlja sprajt pointer. Ako pomenite video bank, onda treba da dodate lokaciju tog banka drugom parametru radi dobijanja prave pozicije. Pointer se postavlja na zadnji multipl broja 64 ako lokacija nije deljiva sa 64.
9.	SYS 49664, „ENA“ s, u	s (0—7) : broj sprajta u (0—1) : uključivanje (0) i isključivanje (1)	Uključuje ili isključuje sprajt.
10.	SYS 49664, „SPR“ s, x, y	s (0—7) : broj sprajta x (0—511) : x koordinata y (0—255) : y koordinata	Postavlja sprajt na zadate x i y koordinate. Treba imati na umu da izvesne koordinate mogu biti van ekrana, pa se sprajt na njima ne vidi.
11.	SYS 49664, „SCO“ s, c, m	s (0—7) : broj sprajta c (0—15) : boja m (0,1) : uključuje ili isključuje multikolor	Postavlja boju sprajta. Poslednji parametar je neobavezan, ali ako je sprajt uključen onda Ø isključuje multikolor za taj sprajt, a 1 ga uključuje.
12.	SYS 49664, „SIZ“ s, x, y	s (0—7) : broj sprajta x (0,1) : y (0,1) : 1 definiše proširenje	Proširuje ili sužava sprajt.
13.	SYS 49664, „PRI“ s, p	s (0—7) : broj sprajta p (0,1) : prioritet : Ø iza teksta, i ispred	Daje prioritet sprajtu, odnosno kazuje računaru da li da stavi sprajt ispred ili iza teksta.
14.	SYS 49664, „SMU“ m_1, m_2	m_1 (0—15) : boja multrkolor 1 m_2 (0—15) : boja multrokor 2	Postavlja multikolore 1 i 2.
15.	SYS 49664, „LSC“ p, k	p (0—24) : početna pozicija za skrolovanje k (0—24):k> p: krajnja za skrolovanje	Skroluje (pomera) ulevo celi sadržaj ili deo ekrana za jedan karakter.
16.	SYS 49664, „RSC“ p, k	isto kao u prethodnoj naredbi	Pomera udesno celi sadržaj ili deo ekrana za jedan karakter.
17.	SYS 49664, „BOR“ c	c (0—15) : boja okvira	Postavlja boju bordera (okvira) ekrana.
18.	SYS 49664, „BAC“ c	c (0—15) : boja pozadine	Postavlja boju pozadine.
19.	SYS 49664, „BLA“ u	u (0,2) : 0 isključuje ekran 1 uključuje ekran	Osvetljava ili potamnjuje ekran. Zatamnjenje se preporučuje pre korišćenja printera 1515 i za ubrzavanje računanja.
20.	SYS 49664, „SCR“ k, r	k (38/40) : broj kolona r (24/25) : broj redova	Uspostavlja širinu i dužinu ekrana.
21.	SYS 49664, „CUR“ x, y	x (0—39) : x koordinata kursora y (0—24) : y koordinata kursora	Daje poziciju kursora.
22.	USR(Ø), „JOY“ p, s	p (0,1): port gde Ø odgovara portu 2 s (0—4) : Ø-gore; 1-dole; 2-levo; 3-desno	Očitava vrednosti džojstika.
23.	SYS 49664, „POS“ x, y	x (0,7) — uobičajeno X=0 y (0,7) — uobičajeno y=3	Uspostavlja sadržaje registara za pomeranje slike. Može se koristiti i za pomeranje celog ekrana.
24.	SYS 49664, „COL“ s, s	s (0,2) — sprajt=0 pozadina=1	Detektuje sudar sprajt/sprajt ili sprajt/pozadina. Fja daje 1 ako je drugi argumenat Ø i ako se sprajt sudara sa drugim sprajtom otkako je Fja korišćena. Takođe vraća 1 ako je drugi argument 1 i ako se sprajt sudario s tekstom od kako se koristi funkcija.

```
1 REM PROGRAM 2
2 REM ******************
3 REM *                *
4 REM *  CBM 64 BASIC  *
5 REM *                *
6 REM *    EXTENSION   *
7 REM *                *
8 REM *  A.STURT  1984 *
9 REM *                *
10 REM******************
80 FS$="C846":REM FREE SPACE STARTS HERE
90 GOTO 5000
100 REM NAMES & ADRESSES
110 DATA VOL,C280,ENV,C295,SIL,C2E4,PUL,C2F2,WAV,C32D,VOI,C460,SOU,C4BB
120 DATA POI,C510,ENA,C538,SPR,C55B,SCO,C590,SIZ,C5CA,PRI,C604,SMU,C629
130 DATA LSC,C685,RSC,C6CD,BOR,C715,BAC,C723,BLA,C731,SCR,C751,POS,C799
140 DATA CUR,C7C6,JOY,C7F2,COL,C81D
290 DATA *
999 REM CODE
1000 DATA 20,73,00,A2,00,20,73,00,9D,A7,02,E8,E0,03,D0,F5,1694
1010 DATA A2,00,BD,00,C0,CD,A7,02,D0,27,BD,01,C0,CD,A8,02,1921
1020 DATA D0,1F,BD,02,C0,CD,A9,02,D0,17,20,73,00,C9,22,D0,1819
1030 DATA F9,8A,4A,AA,BD,00,C1,85,A7,BD,01,C1,85,A8,6C,A7,2272
1040 DATA 00,E8,E8,E8,BD,00,C0,D0,04,E8,4C,12,C2,A2,0B,6C,2090
1050 DATA 00,03,98,48,8A,48,20,86,AE,20,AA,B1,84,A7,85,A8,1756
1060 DATA 68,AA,68,A8,A5,A7,60,86,A9,C5,A9,B0,05,A2,0E,6C,2108
1070 DATA00,03,C0,FF,F0,07,C8,84,A9,C5,A9,B0,F0,60,00,00,2076
1080 DATA20,52,C2,A2,00,A0,0F,20,67,C2,AD,18,D4,29,F0,05,1669
1090 DATAA7,8D,18,D4,60,20,52,C2,A2,01,A0,03,20,67,C2,AA,1773
1100 DATACA,A9,D4,85,FE,A9,05,85,FD,E0,00,F0,0B,18,A5,FD,2447
1110 DATA69,07,85,FD,CA,4C,A9,C2,A0,00,84,02,20,52,C2,A2,1903
1120 DATA00,A0,0F,20,67,C2,0A,0A,0A,0A,8D,A7,02,20,52,C2,1162
1130 DATA A2,00,A0,0F,20,67,C2,0D,A7,02,A4,02,91,FD,C8,C0,1804
1140 DATA02,D0,D7,60,A2,19,A9,00,9D,FF,D3,CA,D0,FA,20,73,2307
1150 DATA00,60,20,52,C2,A2,01,A0,03,20,67,C2,AA,CA,A9,D4,1812
1160 DATA85,FE,A9,02,85,FD,E0,00,F0,0B,18,A5,FD,69,07,85,2106
1170 DATAFD,CA,4C,06,C3,20,52,C2,A9,0F,C5,A8,B0,03,4C,6D,1953
1180 DATAC2,A0,00,A5,A7,91,FD,A5,A8,C8,91,FD,60,20,52,C2,2419
1190 DATAA2,01,A0,03,20,67,C2,85,FD,AA,A9,00,9D,A9,02,20,1740
1200 DATA52,C2,A2,00,A0,03,20,67,C2,AA,A9,10,E0,00,F0,05,1754
1210 DATA0A,CA,4C,4C,C3,A6,FD,9D,A9,02,60,79,00,C9,2C,F0,2008
1220 DATA01,60,20,52,C2,A2,00,A0,01,20,67,C2,85,FE,A9,01,1614
1230 DATAA6,FD,E0,01,F0,05,0A,CA,4C,72,C3,85,FD,A5,FE,F0,2531
1240 DATA 09,AD,17,D4,05,FD,8D,17,D4,60,A9,FF,38,E5,FD,85,2242
1250 DATAFE,2D,17,D4,8D,17,D4,60,00,00,00,00,00,00,00,00,1006
1260 DATA0C,01,1C,01,2D,01,3E,01,51,01,66,01,7B,01,91,01,606
1270 DATAA9,01,C3,01,DD,01,FA,01,18,02,38,02,5A,02,7D,02,1142
1280 DATAA3,02,CC,02,F6,02,23,03,53,03,86,03,BB,03,F4,03,1317
1290 DATA 30,04,70,04,B4,04,FB,04,47,05,98,05,ED,05,47,06,1159
1300 DATAA7,06,0C,07,77,07,E9,07,61,08,E1,08,68,09,F7,09,1265
1310 DATA8F,0A,30,0B,DA,0B,8F,0C,49,0D,18,0E,EF,0E,D2,0F,1203
1320 DATAC3,10,C3,11,D1,12,EF,13,1F,15,60,16,B5,17,1E,19,1337
1330 DATA9C,1A,31,1C,DF,1D,A5,1F,87,21,86,23,A2,25,DF,27,1505
1340 DATA3E,2A,C1,2C,6B,2F,3C,32,39,35,63,38,BE,3B,4B,3F,1257
1350 DATA0F,43,0C,47,45,4B,BF,4F,7D,54,83,59,D6,5E,79,64,1537
1360 DATA73,6A,C7,70,7C,77,97,7E,1E,86,18,8E,8B,96,7E,9F,1956
1370 DATAFA,A8,06,B3,AC,BD,F3,C8,E6,D4,8F,E1,F8,EE,2E,FD,3002
1380 DATA20,52,C2,A2,01,A0,03,20,67,C2,AA,CA,86,FC,A9,D4,2102
1390 DATA85,FE,A9,00,85,FD,E0,00,F0,0B,18,A5,FD,69,07,85,2104
1400 DATAFD,CA,4C,76,C4,20,52,C2,A2,00,A0,60,20,67,C2,C9,2101
1410 DATA00,D0,0A,A0,04,A6,FC,BD,AA,02,91,FD,60,A0,00,AA,1985
1420 DATA CA,8A,0A,AA,BD,A0,C3,91,FD,BD,A1,C3,C8,91,FD,A6,2771
1430 DATAFC,BD,AA,02,A0,04,09,01,91,FD,60,20,60,C4,20,52,1719
1440 DATAC2,A5,A7,D0,07,A5,A8,D0,03,4C,93,C4,38,A5,A7,E9,2325
1450 DATA01,85,A7,A5,A8,E9,00,85,A8,A2,F9,A0,00,C8,D0,FD,2400
1460 DATAE8,D0,FA,20,E1,FF,F0,03,4C,C1,C4,60,48,A9,01,E0,2472
1470 DATA00,F0,05,0A,CA,4C,EF,C4,AA,A0,00,68,F0,06,8A,11,1803
1480 DATAFD,91,FD,60,86,FB,38,A9,FF,E5,FB,31,FD,91,FD,60,2888
1490 DATA20,52,C2,A2,00,A0,07,20,67,C2,85,FB,20,52,C2,A5,1823
1500 DATAA6,A2,00,A0,3F,20,67,C2,06,97,26,A8,06,A7,26,A8,1540
1510 DATA25,A6,A6,FB,9D,F8,07,60,20,52,C2,A2,00,A0,07,20,1927
1520 DATA67,C2,85,FB,20,52,C2,A2,00,A0,01,20,67,C2,A6,FB,2058
1530 DATAA0,15,84,FD,A0,D0,84,FE,4C,EC,C4,20,52,C2,A2,00,2298
1540 DATAA0,07,20,67,C2,85,FC,20,52,C2,A5,A8,A2,00,A0,01,1845
1550 DATA20,67,C2,A6,FC,A0,10,84,FD,A0,D0,84,FE,20,EC,C4,2526
1560 DATA06,FC,A5,A7,A6,FC,9D,00,D0,20,52,C2,9D,01,D0,60,2143
1570 DATA20,52,C2,A2,00,A0,07,20,67,C2,85,FB,20,52,C2,A2,1820
1580 DATA00,A0,0F,20,67,C2,A6,FB,9D,27,D0,20,79,00,C9,2C,1723
1590 DATAF0,01,60,20,52,C2,A2,00,A0,01,20,67,C2,A6,FB,A0,1874
1600 DATAD0,84,FE,A0,1C,84,FD,4C,EC,C4,20,52,C2,A2,00,A0,2305
1610 DATA07,20,67,C2,85,02,20,52,C2,A2,00,A0,01,20,67,C2,1431
1620 DATAA6,02,A0,D0,84,FE,A0,1D,84,FD,20,EC,C4,20,52,C2,2269
1630 DATAA2,00,A0,01,20,67,C2,A0,D0,84,FE,A0,17,84,FD,A6,2140
1640 DATA02,4C,EC,C4,20,52,C2,A2,00,A0,07,20,67,C2,85,FB,1969
1650 DATA20,52,C2,A2,00,A0,01,20,67,C2,A6,FB,A0,D0,84,FE,2131
1660 DATAA0,1D,84,FD,49,01,4C,EC,C4,A0,00,84,02,20,52,C2,1756
1670 DATAA2,00,A0,0F,20,67,C2,A6,02,9D,25,D0,E8,86,02,E0,1828
1680 DATA02,D0,EA,60,20,52,C2,A2,00,A0,18,20,67,C2,8D,A7,1831
1690 DATA02,20,52,C2,A2,00,A0,18,20,67,C2,8D,A8,02,CD,A7,1668
1700 DATA02,B0,03,4C,6D,C2,60,A9,00,85,FB,A9,04,85,FC,C9,1959
1710 DATA00,D0,01,60,18,A5,FB,63,28,85,FB,A5,FC,69,00,85,1929
1720 DATAFC,88,4C,6F,C6,20,44,C6,AC,A7,02,20,67,C6,AE,A7,2086
1730 DATA02,A0,01,B1,FB,88,91,FB,C8,18,A5,FC,69,D4,85,FC,2466
1740 DATAB1,FB,88,91,FB,38,A5,FC,E9,D4,85,FC,C8,C8,C0,28,2895
1750 DATAD0,E1,88,A9,20,91,FB,18,A5,FB,69,28,85,FB,A5,FC,2552
1760 DATA69,00,85,FC,E8,EC,A8,02,90,C7,F0,C5,60,20,44,C6,2302
1770 DATAAC,A7,02,20,67,C6,AE,A7,02,A0,26,B1,FB,C8,91,FB,2239
1780 DATA88,18,A5,FC,69,D4,85,FC,B1,FB,C8,91,FB,38,A5,FC,2776
1790 DATAE9,D4,85,FC,88,88,C0,FF,D0,E1,C8,A9,20,91,FB,18,2803
1800 DATA85,FB,69,28,85,FB,A5,FC,69,00,85,FC,E8,EC,A8,02,2490
1810 DATA90,C7,F0,C5,60,20,52,C2,A2,00,A0,0F,20,67,C2,8D,1991
1820 DATA20,D0,60,20,52,C2,A2,00,A0,0F,20,67,C2,8D,21,D0,1692
1830 DATA60,20,52,C2,A2,00,A0,01,20,67,C2,C9,00,F0,09,AD,1679
1840 DATA11,D0,29,EF,8D,11,D0,60,AD,11,D0,09,10,8D,11,D0,1756
1850 DATA60,20,52,C2,A2,26,A0,28,20,67,C2,C9,27,D0,03,4C,1660
1860 DATA6D,C2,C9,26,D0,0B,AD,16,D0,29,F7,8D,16,D0,4C,79,2020
1870 DATAC7,AD,16,D0,09,08,8D,16,D0,20,52,C2,A2,18,A0,19,1669
1880 DATA20,67,C2,C9,18,D0,09,AD,11,D0,29,F7,8D,11,D0,60,1919
1890 DATAAD,11,D0,09,08,8D,11,D0,60,20,52,C2,A2,00,A0,07,1514
1900 DATA20,67,C2,85,FB,AD,16,D0,29,F8,05,FB,8D,16,D0,20,2064
1910 DATA52,C2,A2,00,A0,07,20,67,C2,85,FB,AD,11,D0,29,F8,2005
1920 DATA05,FB,8D,11,D0,60,20,52,C2,A2,00,A0,27,20,67,C2,1716
1930 DATA85,02,20,52,C2,A2,00,A0,18,20,67,C2,AA,A4,02,4C,1530
1940 DATAF0,FF,A0,00,B1,FD,E0,00,F0,05,4A,CA,4C,E6,C7,29,2376
1950 DATA01,60,20,52,C2,A2,00,A0,01,20,67,C2,48,20,52,C2,1487
1960 DATAA2,00,A0,04,20,67,C2,AA,68,A8,A9,DC,85,FE,A9,00,2042
1970 DATA85,FD,20,E4,C7,49,01,A8,A9,00,6C,05,00,20,52,C2,1677
1980 DATAA2,00,A0,07,20,67,C2,48,20,52,C2,A2,00,A0,01,20,1393
1990 DATA67,C2,A8,68,AA,A9,D0,85,FE,A9,1E,85,FD,20,E4,C7,2547
2000 DATAA8,A9,00,6C,05,00,AF,A4,AC,8C,FF,84,FE,8A,EF,20,2151
2010 DATA*
2999 REM STORE STATEMENT NAMES & ADRESSES
3000 A1=49152:A2=49408
3010 READA$:IFA$="*"THEN3050
3020 FORN=1TO3:POKEA1-1+N,ASC(MID$(A$,N,1)):NEXT:POKEA1+3,0
3030 READH$:GOSUB20000:POKEA2+1,INT(D/256):POKEA2,D-PEEK(A2+1)*256
3040 A1=A1+4:A2=A2+2:GOTO3010
3050 POKEA1-1,255:RETURN
3999 REM STORE CODE
4000 A=49664
4010 CS=0:READH$:PRINTH$;:IFH$="*"THEN RETURN
4020 GOSUB20000:PRINTD:CS=CS+D:POKEA,D:A=A+1
4030 FORN=1TO15:READH$:PRINTH$;:GOSUB20000:PRINT" ";D
4040 POKEA,D:A=A+1:CS=CS+D:NEXT:READC:PRINTC,CS
4050 IFC<>CSTHEN PRINT"DATA ERROR IN"PEEK(63)+256*PEEK(64):END
4060 GOTO4010
5000 PRINT"■■■■■CBM 64 EXTENSION"
5010 PRINT"■LOADING DATA.  PLEASE WAIT."
5020 GOSUB3000
5030 GOSUB4000
5040 PRINT"GET CODE TAPE READY,THEN PRESS A KEY....":POKE198,0:WAIT198,1
5045 A$="EXTENSION 64"
5050 FORN=1TO12:POKE678+N,ASC(MID$(A$,N,1)):NEXT
5060 POKE 780,12:POKE781,167:POKE782,2
5070 SYS65469
5080 POKE253,0:POKE254,192:POKE780,253
5090 H$=FS$:GOSUB20000
5100 POKE782,INT(D/256):POKE781,D-PEEK(782)*256
5110 SYS65486
5999 END
20000 REM CONVERT HEX-DEC
20010 D=0
20020 IFH$=""THEN FOR I=1TOLEN(H$):J=ASC(MID$(H$,I,1))-48:D=D*16+J+(J>9)*7
20030 IFH$=""THEN NEXT
20040 RETURN
```

odgovarajuća dejstva od adresa 49408. Za svako ime ostavljena su po četiri registra, što znači da imate prostora za 64 nove instrukcije (ovim programom uvedene su 24). U prva tri registra smeštaju se kodovi slova iz imena, a u četvrtom registru sve nule sem za poslednji podatak kad se postavljuju sve jedinice. Ovo smeštanje podataka vrši se potprogramom 3000 koji konverziju adresa iz azbučnog u numerički oblik vrši uz pomoć potprograma 20000. Zatim se prelazi na postavljanje mašinskih rutina počev od adrese 49664. Ovo smeštanje uz proveru da li ste korektno uneli podatke vrši potprogram 4000. On, takođe, poziva potprogram 20000 za konverziju podataka. (Kada je pozivan iz potprograma 3000, ovaj potprogram je radio sa četvorocifrenim heksadecimalnim brojevima, a u ovom slučaju radi sa dvocifrenim).

Na kraju, kao što smo već ranije rekli, nailazi deo programa koji ove mašinske rutine snima na traku pod nazivom „Extension 64".

Dejstvo novih naredbi ilustruje program 3 koji ih koristi. Ukoliko niste sigurni u način njihove upotrebe, pogledajte listing. Da biste dobili još komandi, prvo učitajte postojeći novi set naredbi, a zatim učitajte program 2. Napišite rutine na više komande, pa ih upišite između adresa $C846 (koja se nalazi u FS$ i $CFFF. U programu 2, između linija 14 Ø i 29 Ø dodajte, u obliku data liste, prva tri slova naredbe i adrese mašinskih programa koji im odgovaraju.

Primer:
naredba „HIR", rutina na adresi $C846
naredba „SET", rutina na adresi $C856
15 Ø DATA HIR, C846, SET, C856

Sledeći potprogrami mogu nam biti od koristi.

$C252: uzima ceo šesnaestobitni broj iz bejzika i stavlja ga u $A7 i $A8, sa najmlađim bajtom u akumulatoru.

$C267: proverava da li je broj koji se nalazi u akumulatoru veći ili jednak broju u X-registru ili manji ili jednak broju u Y-registru. Ako nije, rutina će naznačiti da se radi o grešci i napisaće „ILLEGAL QUANTITY ERROR"

$C4EC: stavlja broj iz akumulatora, koji mora biti Ø ili 1, u bit X lokacije u $FD i $FE.

$C7E4: stavlja bit X lokacije sadržane u $FD i $FE u akumulator.

Kada sve ovo budete uradili, izmenite liniju 8Ø tako da FS$ sadrži adresu sledećeg slobodnog bajta posle vaše rutine. Pazite da broj vaših novih komandi ne bude veći od 64. Na kraju, otkucajte u direktnom modu:
FS$ = „(početak slobodne memorije, heksadecimalno)":
GOSUB 3ØØØ: GOTO 5Ø 4Ø
Podaci će tada biti sačuvani.

Nevenka Špalević

Mali oglasi

Prvog novembra stupile su na snagu nove cene malih oglasa u časopisu „Galaksija" i specijalnom izdanju „Računari". Kao i do sada, mali oglasi su podeljeni na dve kategorije: klasične male oglase i male oglase u okviru.

Klasični mali oglasi

Mali oglasi do dvadeset reči koštaju 400 dinara. Oglas ne sme biti duži od pedeset reči. Dvadeset i prva i svaka dalja reč do pedesete košta po 20 dinara. Adresa oglašivača se ne računa u cenu.

Prva reč u malom treba da odražava njegovu suštinu. Mali oglasi, dakle, treba da počinju sledećim rečima **Prodajem, Kupujem, Držim časove, Menjam . . .**

Mali oglasi u okviru

Cena uokrirenog malog oglasa utvrđuje se na osnovu prostora koji oglas zauzima na jednom „galaksijinom" stupcu. Jedan visinski centimetar košta 300 dinara, s tim što se mogu zakupiti najmanje dva centimetra. Na jedan visinski centimar staju četiri reda sa po 32 slovna znaka. Oglašivač može ali ne mora da iskoristi čitav prostor u jednom redu. U tom slučaju, računa se broj redova a ne broj slovnih znakova.

Kategorije

Uz svaki mali oglas, klasičan ili u okviru, treba navesti jednu od sledećih kategorija sa njenom podkategorijom:

ZX 81	• QL	BBC	DŽEPNI RAČUNARI
• hardver	• hardver	• hardver	• hardver
• softver	• softver	• softver	• softver
• knjige	• knjige	• knjige	• knjige
„SPEKTRUM"	„KOMODOR"	„ELEKTRON"	„GALAKSIJA"
• hardver	• hardver	• hardver	• hardver
• softver	• softver	• softver	• softver
• knjige	• knjige	• knjige	• knjige

Popusti

Oglasi za prodaju originalnih programa za računar „galaksija" objavljuju se besplatno, pod uslovom da nemaju više od dvadeset reči. Za duže oglase treba doplatiti samo razliku u ceni. Na oglase za prodaju originalnih programa za sve ostale kućne računare odobravamo popust od 25% pod uslovom da oglašivač uveri redakciju u autorstvo programa.

Uplata

Prijem malih oglasa se zaključuje 35 dana pre izlaska „galaksije" odnosno „Računara" iz štampe. Uplata se vrši poštanskom uplatnicom na adresu redakcije („Galaksija", Bulevar vojvode Mišića 17, 11000 Beograd, „za male oglase") ili na naš žiro-račun (vidi treću stranu). Uz tekst oglasa, treba obavezno dostaviti i potvrdu o uplati. Oglasi koji ne poštuju navedene propozicije (na primer, zahtevaju veći prostor od onoga za koji je uplaćeno) neće biti uvršćeni bez doplate.

- Spectrum programi: igre 50 d. sistemski i uslužni programi veliki izbor najnovijih igara, kvalitet, najniže cene. Uskoro Galaksija Software! Pozivi LSB software. Tel: 418-908
- Najjeftiniji programi za Spectruma (16 i 48K)!!! Od preko 150 najboljih programa izaberite 30 po sopstvenoj želji, za koje ćete platiti samo 1000 n.d. Spisak programa tražite na adresu Dakić Jovan, Bulevar revolucije 420/40, Beograd, tel. 011/414-997 ili 414-862.
- NOVO! Neophodni hardverski dodatak za vaš kompjuter. MOODE 1 omogućava razmenu programa i podataka putem telefona. Radi sa svim kompjuterima: SP ECTRUM, ZX81, COMMODORE, GALAKSIJA, itd. Jednostavna upotreba, siguran rad, baterijsko napajanje, profesionalni kvalitet izrade. Isporuka pouzećem za svega 5000 dinara. PNP Electronics. Jeretova 12, 58000 SPLIT.
- SPECTRUM, igre, business, korisni programi, uputstva — ima skoro sve. Pišite — dobićete popis. Reuf Koluder, M. Tita 32, 71000 Sarajevo (071) 23-043.
- **Prodajem** Commodore 64 potpuno nov za 85000 din. Tel. 011/4443-351.

NOVO
COMETSOFTWARE
stručna i popularna literatura na engleskom jeziku
iz oblasti računara i računarske tehnike
- mikroprocesori Z80, 80/86, 6502, 6509, Z8000, 68000
- programiranje u bejziku i mašinskom jeziku za računare „spectrum", „komodor 64" i BBC
- servisni priručnici za „komodor 64" i „epson"

Ljubiša Milovanović
Petra Lekovića 57
11030 Beograd
Tel.: 011/558-007

SPECTRUM

Najnoviji programi direktno iz Londona: Šerlok Holms, World Cup Football, TII, Tenis i još 300 drugih programa koji se nalaze u besplatnom katalogu. Niske cene uz upustva za sve programe i kvalitetne kasete. Jeremić Nebojša, Risanska 10, 11000 Beograd, tel. 011/643-061.

- **Eprom servis u Zagrebu.** Izmjene listinga i 25XX 27XX presnimavanje za Apple, „galaksiju", AIM65 Spectrum ULA. Isporuka odmah. Novo! Adria 48 kit (Apple kompatibilno). Landeka, Domagojeva 17, Zagreb, tel. 041/529-586 od 19 časova.

Byte Shop
The best software shop all over the world presents you more then 300 programs for you ZX Spectrum, including clasic arcade, adventure games but also the last software hits in U. K. Specialy offer: cassette with 50 basic games only 1500 din. Write or call us and we'll do it for you quickly, sheeper then other co.
Prices:
— 16K — 60 din.
— 48K — 100 din.
Please, send 50 din. for catalogue with games description. We want you to play the best!!!
Address: Omrčen Damir, Bulevar JNA 64/10, Beograd, phone: 011/662-044

• Vlasnici Spectruma, ne propustite.

COMET SOFTWARE
Najveći izbor programa i literature za vaš ZX SPECTRUM
— video igre
— logičke igre
— namenski programi
— knjige
— uputstva za programe
Programe snimamo DIREKTNO IZ SPECTRUMA!
COMET software vam jedini nudi programe sa GARANCIJOM!!
Snimamo programe po vašem izboru!
Imamo i pakete programa koji su znatno jeftiniji!
Tražite besplatni katalog!
Za sve što vas interesuje obratite se na adresu:
MILOVANOVIĆ LJUBIŠA, Petra Lekovića 57, 11030 BEOGRAD, tel: 011/558 007

- **Spektrumovci!** Javite se ako su vam nepoznati hit programi: Decatlon, Zombie, Underworlde, Strangeloop, Montymole . . . Možete naručiti i ostalih 200 programa. Katalog 50 din. Zvonko Šošević, Radmile Rajković 10, 11000 Beograd, 011/767-398.
- **Za ZX Spectrum** oko 500 programa. Pojedinačna cena 40 dinara ili u polusatnim kompletima po 160 dinara. Katalog košta 20 din. Marko Marković, Dž. Bijedića 27 A/XI, 71000 Sarajevo.

Za Commodore — 64 više od 500 najboljih programa!!!
Pri većim narudžbinama specijalni popust. Najminimalnije cene i besplatan katalog. Svi programi biće direktno snimljeni i verifikovani iz CBM—64. Možda niste svesni mogućnosti Simon's Basic-a ili pak ne vladate nemačkim jezikom. Evo prilike — za svega 1400 din. — naručite prevod istoimene knjige. Osim ovoga, možete nabaviti i literaturu za vaš CBM—64, od Basic-a disasembinga ROM-a na nemačkom i engleskom jeziku. Predrag Bogdanović, 7. jula 36/III, 11000 Beograd, tel. 011/636-328.

- **CBM 64**
Prodajem i menjam više od 650 programa. Igre (arkadne, avanture, simulatori . . .), Utility PGMI (grafika, muzika . . .), aplikativni PGMI (tekst procesori, database . . .), sistemski PGMi (kompileri, Simon's Basic, Pascal . . .). Niske cene — novi programi. Tražite katalog programa za Commodore 64. Sodnik/Korošec, C. Ceneta Štuparja 3, 61.231 Črnuče — Ljubljana, tel. 061/373-138.

Commodore Royal Service- Coros. Ne propustite priliku da veoma jeftino obogatite vašu kolekciju programa za CBM—64! Šta oblikuje Coros? — brza isporuka (1—3 dana), pristupačne cene (500—100 din.), 100% sigurnost snimka, besplatan katalog, profesionalni kvalitet.
We want you to play the best on your CBM—64!!! Omrčen Damir, Bulevar JNA 64/10, 11000 Beograd, tel. 011/662-044.

- **COMMODORE 64-** Flight Simulator II, Practifile, Superbase, Wiza Write, Doctor 64, Sim. Basic Extewded, Mae, Kolara, Synthimat i ostali biseri svetske produkcije -100 din. po programu. Izbor iz 700 programa, literatura i manuali. tražite bliže informacije i besplatan katalog. Veselin Milisavljević, Vitanovačka 42, stan 45, Beograd, tel. 011/462-659.
- **„RASUMI TO RADI KOMPLEKSNO!** Za vaš C—64 programi za obradu podataka, igre, utility programi, literatura i prijevodi, nudi Computerlab „RASUMI", 54103 Osijek, Poštanski fah 313.
- **COMMODORE 64-** Kompletna ponuda stručno prevedene orginalne literature (offset štampa, latinica)
— prevod uputstva i basica (manual)
— prevod uputstva za „Simons Basic" i „Pascal".
Naručite pouzećem. Liber Commodore, Osijek, Drinska 91, tel. 054/25-513.
- **Naručite** za Commodore 64 i drajv hit program u SAD: Flight Simulator II. Kopisanje sa jedinog orginala u zemlji. Tel. 011/156-444.

JOYSTICK CLUB
software
ZX SPECTRUM ultra-software.
YU-premijere najnovijih hitova.
Da li ste već postali član JOYSTICK CLUBA?
Vladimir Milićević
Jovana Popovića 19A, Beograd tel. 460-128
Stevan Milićević
Gogoljeva 44. Beograd
tel. 550-972

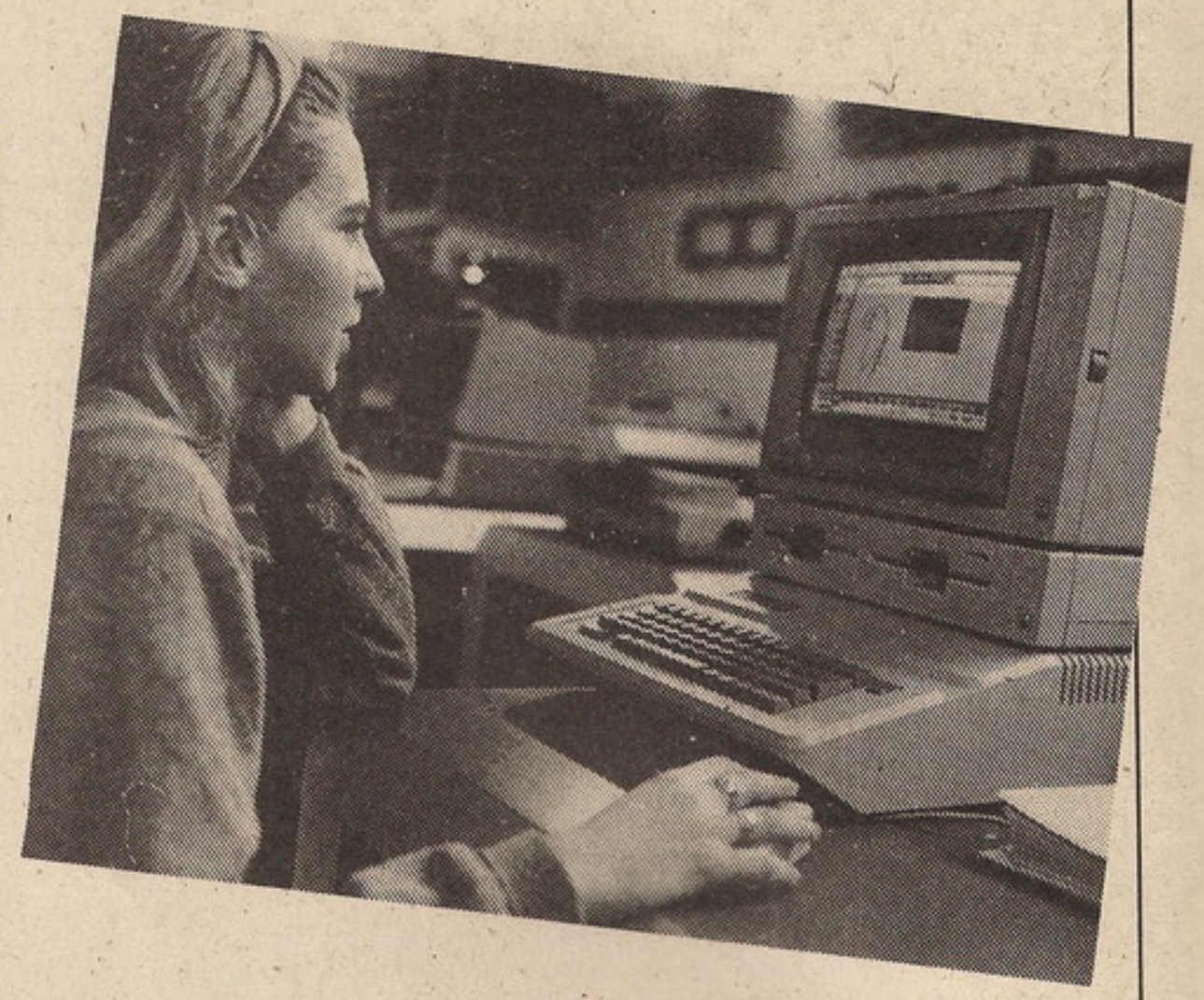
Comodore
Comodore
PROGRAMI ZA COMODOR 64
JEDINSTVENE CENE,
KVALITETNI SNIMCI,
UPUTSTVA, KATALOG
NENAD JEREMIĆ-
RISANSKA 1o
11000 BEOGRAD 643-061
64

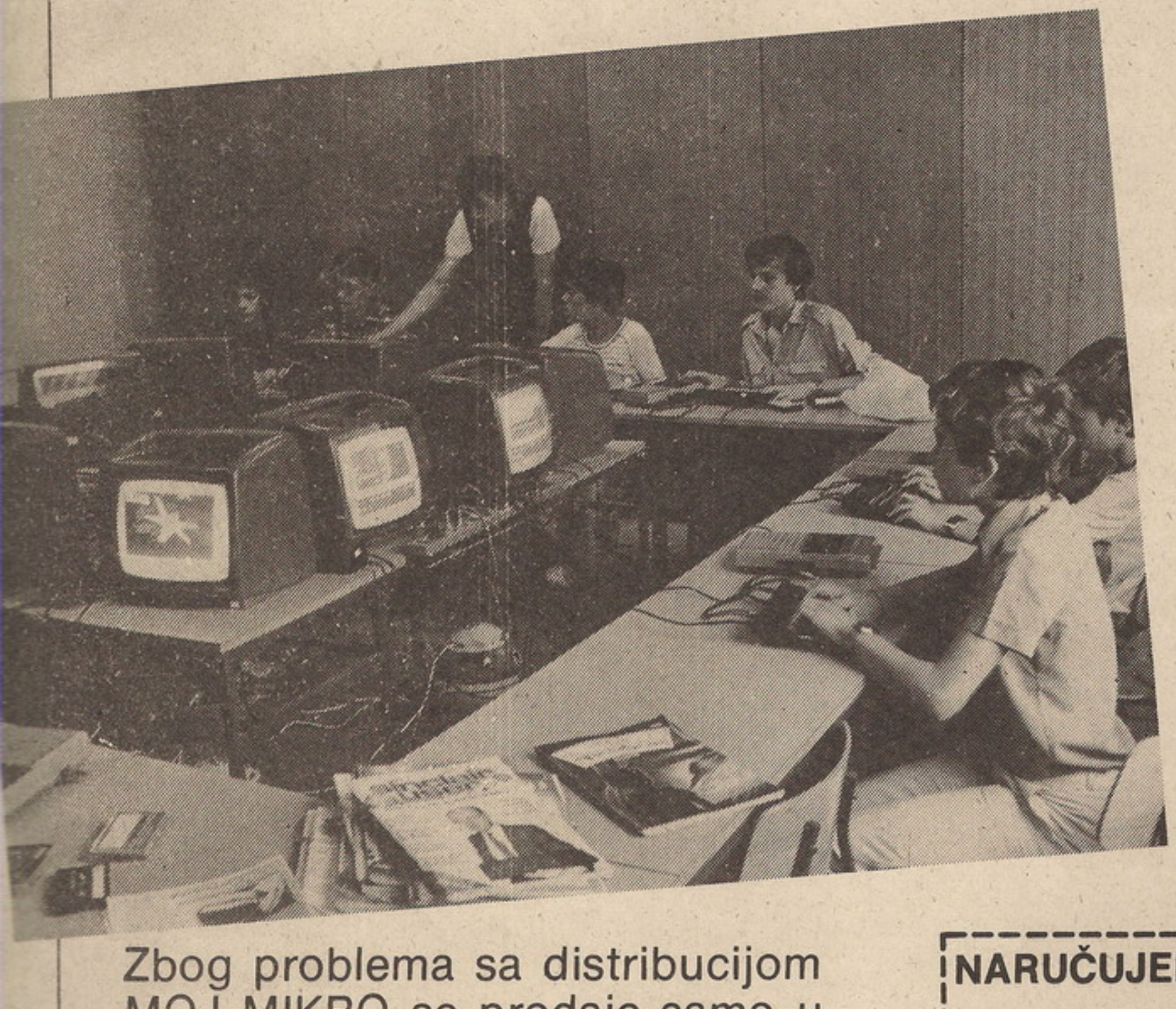
Upoznajte

MOJ MIKRO

YU revija za hekere

Posle velikog uspeha kod slovenačke publike MOJ MIKRO sada i na srpskohrvatskom jeziku.
MJ MIKRO je rezultat saradnje revije Teleks i softverske redakcije Radio Študent iz Ljubljane.
Prvi broj izašao je sredinom decembra i zapravo je zbir najboljeg iz dosadašnjih pet slovenačkih brojeva.





Iz sadržaja: — Originalni supertest — QL — Testovi — Macintosh, HP 150, Gemini — Čudesni svet dodataka — Crtamo sa kompjuterom Prilog originalnih programskih listinga (16 stranica) Hardverski saveti — posebno za YU hekere Kako proširiti memoriju Spectruma? Naučna fantastika

Zbog problema sa distribucijom MOJ MIKRO se prodaje samo u većim gradovima. Zbog toga vam savetujemo da se pretplatite pomoću narudžbenice.
Narudžbenicu šaljite na adresu: MOJ MIKRO, ČGP Delo, oddelek naročnin, Titova 35, 61000 Ljubljana

NARUČUJEM MOJ MIKRO

Pretplatu ću poslati kad primim uplatnicu

— — — — — — — — — — — — —
(ime i prezime) (ulica, kućni broj)

— — — — — — — — — — — — —
(broj pošte) (pošta)

— — — — — — — — — — — — —
(potpis)

Knjiga koja vas popularno uvodi u svijet kompjutora!

Peter Laurie

KOMPJUTOR U KUĆI

priručna enciklopedija

Najtraženija engleska knjiga o kompjutorima — sada i na našem jeziku!

Iz sadržaja: — OPĆENITO O KOMPJUTORU — PROGRAMIRANJE — PRIMJENA U PRAKSI: Poslovni programi, kompjutor i slikarstvo, simulacije, manipulacija slike, kompjutorska animacija, kompjutor koji govori, kompjutorska glazba, robotika, roboti u industriji, programiranje prilagođeno za kućne kompjutore SINCLAIR itd. — POGLED U BUDUĆNOST — RJEČNIK POJMOVA — SHEMATSKE SKICE I UPUTE itd.

Temeljni priručnik za svakog koji namjerava nabaviti kompjutor za kuću, školu ili za poslovne potrebe

Bitne osobine: jasnoća i preglednost teksta, stručna pouzdanost objašnjenja i bogatstvo korisnog ilustrativnog materijala

Pretplatna cijena: 2.200.- dinara
Prodajna cijena: oko 3.000.- dinara
Izlazi iz tiska: 24. oktobra 1984.

Narudžbenica Računari III

Izdavačkoj jedinici „Cankarjeve založbe"
Ilica 26, 41000 Zagreb

Ovime naručujem _____ primjeraka knjige KOMPJUTOR U KUĆI (izdanje na srpskohrvatskom) po pretplatnoj cijeni od 2.200.- dinara. što ću platiti u pet mjesečnih rata po 440.- dinara.

Knjigu ću primiti po izlasku iz tiska i uplati cjelokupne pretplatne cijene.

Ime i prezime naručitelja .....

Adresa .....

.....

.....

Vlastoručni potpis

Napomena: pretplatiti se možete i na telefonski broj (041) — 432-325

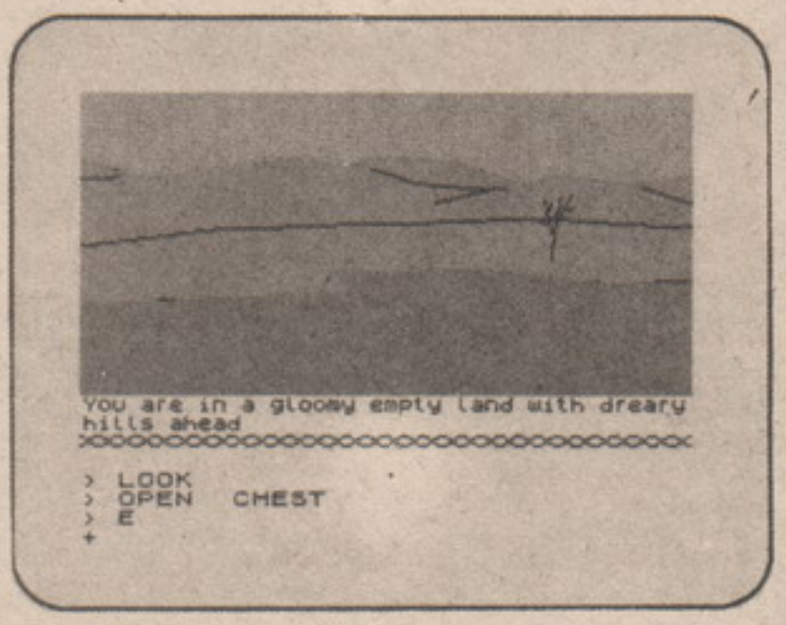
Katalog najboljih igara svih vremena za „spektrum“

PEDESET VELIČANSTVENIH

Urednički tim engleskog mesečnika „Sinclair User“ upustio se nedavno u, kako je neko od njih rekao, „samoubilački“ posao — izbor pedeset najboljih igara za „spektrum“. Da bi se neka igra našla na tom spisku „veličanstvenih“, ona uopšte nije morala da bude bestseler; doista, neke izabrane igre postigle su sasvim skroman komercijalni uspeh. Od nje se, zapravo, tražilo da predstavlja svojevrsni kamen-međaš na lavirintskom putu kojim se industrija igara kreće poslednjih godina.

„Pedeset veličanstvenih“ je vodič do onog najboljeg u softveru za „spektrum“, ali isto tako, i kondenzovana istorija neprekidnog razvoja programiranja igara. Ovaj spisak najkvalitetnijih programa, razume se, obuhvatio je samo igre, a ne i obrazovanje, posao, uslužne primene, praktičnu upotrebu i sistemski softver — gde je vrednovanje mnogo više relativizirano vlastitim potrebama kosnika.

Moguće je da se pojedini čitaoci „Računara“ neće složiti sa izborom koji su načinili urednici časopisa „Sinclair User“ — sa njihovom selekcijom programa koje bi svaki kolekcionar igara trebalo da ima. Ipak, treba da imaju u vidu da su kolege iz Engleske svoj posao obavile korektno. To potvrđuje i njihovo priznanje da su u nekoliko mahova bili na ivici da se potuku — svaki put kada bi sa spiska ispala omiljena igra nekog od članova uredničkog žirija.



1 THE HOBBIT — Hobit

Melbourne House

Ljubitelji kompjuterskih igara dele se u dve kategorije: na one koji su uspeli da pobegnu iz zloduhove podzemne tamnice i one koji još uvek drežde u njoj.

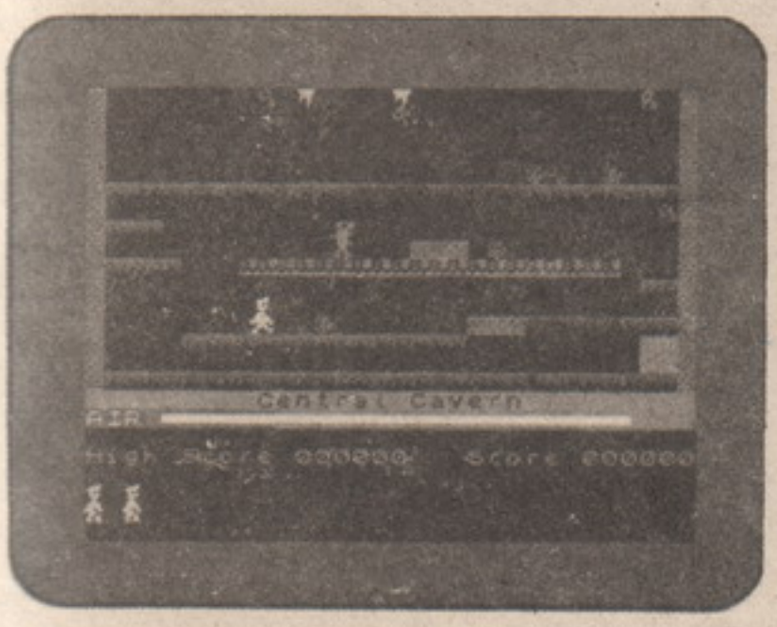
Kada je razvijen 1982. godine, „Hobit“ je bio ispred svoga vremena, i ostao je igra avanture sa kojom se sve druge porede. Zadatak je da sledite stope Tolkinovog Bibo Baginsa, koji traga za blagom zmaja Smauga. Igrom se upravlja rečima i rečenicama na modifikovanom engleskom jeziku. Zaplet se menja svaki put kada nešto odigrate, a spisak mogućih rešenja i složenost igre sve više rastu.

2 LORDS OF MIDNIGHT — Gospodari ponoći

Beyond

Bez sumnje jedna od najneobičnijih igara za „spektrum“, „Gospodari ponoći“ su avanturistička ratna igra širokih razmera. Izvanrednim programiranjem obezbeđeno je čak 32.000 lokacija na ekranu!

Vaš cilj je da, kao Luksor Princ Meseca, porazite armije Zlomraka, bilo ratom bilo krađom i uništenjem Ledene krune. Za tu svrhu moguće je regrutovati mnogobrojna lica i snage, a mnoštvo mogućih lokacija i taktika pružaju igraču beskrajno zadovoljstvo. Ova igra uvodi vas u svet magije i straha i ima moć da uzbudi i inspiriše više nego ijedna druga.

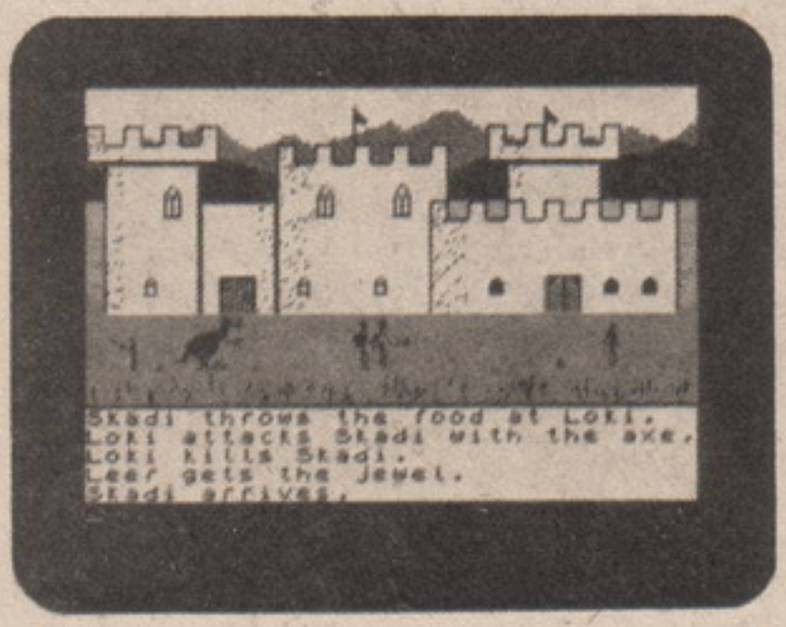


3 MANIC MINER — Rudar manijak

Software Projects

Adaptacija ove klasične akcione igre priređena za „spektrum“ pokazala se kao krajnji domet u toj vrsti igara i podstakla je druge kompanije da proizvode sličan softver.

Vaš zadatak je da se probijete kroz što veći broj podzemnih nivoa da biste prikupili ključeve rudnika u kome radi Vili. Primorani ste da bežite od pingvina, mutantnih toaleta, otrovnih mačuhica i rudarskih robota. Na kraju možete da pristupite specijalnom nivou od opakih slatkiša koji su u stanju da dokrajče svakog rudara.

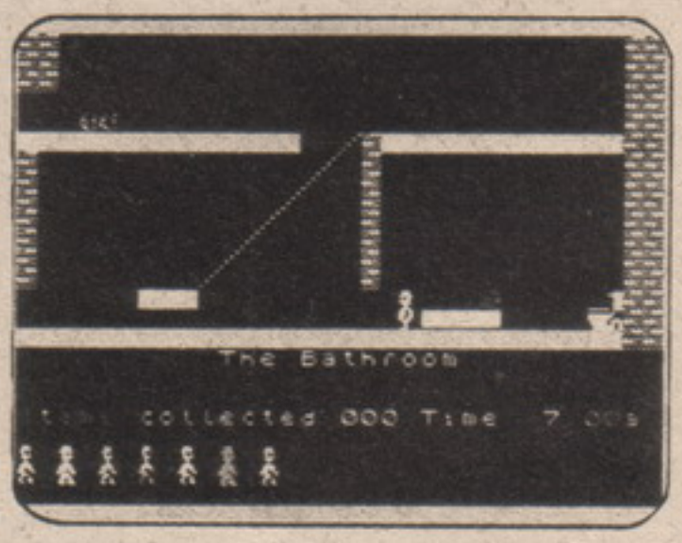


4 VALHALLA — Valhala

Legend

Posle „Hobita“, prvi pravi prodor u avanturama napravila je „Valhala“, igra koja je uspostavila nove standarde u interakciji grafike i teksta.

Vi ste malo norveško božanstvo i morate se upustiti u osvajanje naročitih ritualnih predmeta. U to je uključeno čitavo mnoštvo bogova, boginja, džinova, patuljaka i sličnih stvorova, koji su na ekranu predstavljeni kao pokretni likovi što se bore, jedu i piju. Svaki lik može da pruža potporu dobru ili zlu, sa različitim stepenima inteligencije i snage. Da biste nadvladali tu potporu morate je uveriti u svoje dobre — ili loše — namere. Usavršen interpretator teksta i moćan zaplet zaokružuju ovu najneobičniju igru.

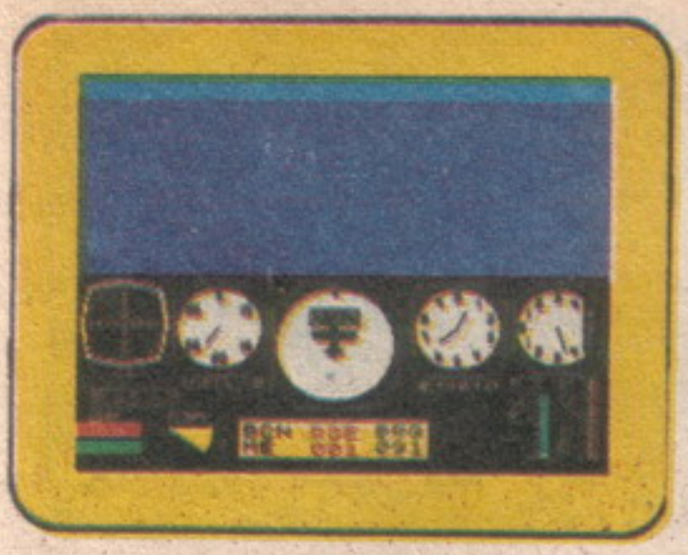


5 JET SET WILLY — Bogataš Vili

Software Projects

Igra je usledila posle klasičnog „Rudara manijaka“ i doživela brz i neočekivan uspeh. Rudar Vili je otišao u penziju i počeo da se opija. Njegova gazdarica Marija ne pušta ga u krevet dok mu ne raskloni flaše i čaše koje okružuju kuću.

Grafika u stilu takozvanih arkadnih igara, velik broj soba i neobična stvorenja koja u njima stanuju obezbeđuju „bogatašu Viliju“ važno mesto u istoriji akcionih igara. Igra je, takođe, poznata po specijalnom sistemu za zaštitu od piratstva — sa, na žalost, ograničenim uspehom.



6 FLIGHT SIMULATION
Psion

Simulacija letenja

Igra je načinila novi prodor 1982. godine, kao prva „ozbiljna" igra za „spektrum". Probu vremena izdržala je krajnje dobro, podstakavši druge kompanije da proizvodu mnoštvo sličnih simulatora.

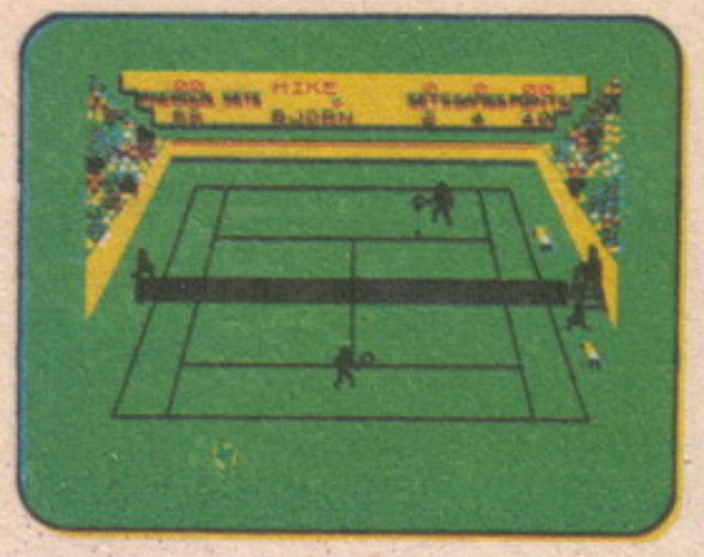
U „Simulaciji letenja" vozite jedan lagani avion sa elisnim pogonom, sposoban za akrobacije ali pod uslovom da ste vešt navigator, naročito na vetru. Velika mapa preko koje možete da letite sadrži i takve detalje kao što su jezera. Mada grafika danas deluje zastarelo, zadovoljstvo zbog upravljanja avionom, makar i na kompjuteru, toliko je da se ljudi uvek iznova vraćaju ovoj igri.

7 ATIC ATAC
Ultimate

Atik atak

Smešten u gotskom zamku sa podzemnim tamnicama i pećinama ispod njih, „Atik atak" je divna akciona igra. Vi se krećete iz sobe u sobu prikupljajući delove zlatnog ključa. Mnoštvo čudovišta trudi se da vas zaustavi, ali postoje i predmeti koji vam pomažu u pohodu.

Igra bi i inače bila lepa, ali pažnja kojom je program urađen čini je izvanrednom. Igrač može da bira između tri lika u igri, čija snaga po efektu pruža mogućnost tri igre. Pažnja kojom su rađeni detalji doista je primerna, sa duhovitim detaljima, kao što je indikator pečenih pilića.

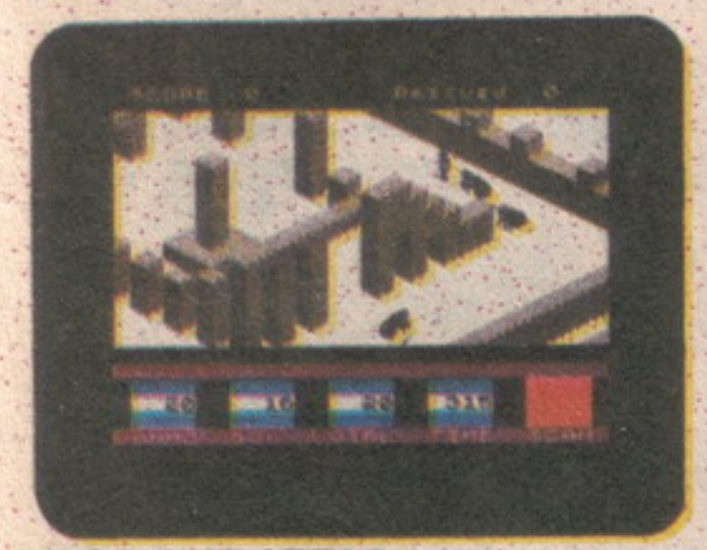


8 MATCH POINT
Psion

Poen u meču

Ovo je jedina igra koja prilično uspešno simulira takmičenje na teniskom igralištu u Vimbldonu. U igri mogu da učestvuju dvojica, ili jedan igrač protiv kompjutera. Tri nivoa složenosti obezbeđuju da igrači mogu da se zagreju u prvoj rundi, a zatim sebi daju oduška do finala.

Najupečatljivija osobina „Poena u meču" je pedantnost kojom je igralište prikazano na ekranu, sve do senke loptice. Kretanje igrača i loptice je ravnomerno i brzo, a akcija je tako realistična da možete pomoću reketa da završite lopticu.



9 3D ANT ATTACK
Quicksilva

Napad mrava

U ovoj akcionoj igri, koja je nagovestila revolucionarnu tehniku nazvanu „trodimenzionalni softsolid" — prikladan naziv za pozadinski scenario — u tri dimenzije se prikazuju grad zvani Antešer, mravi i junak i junakinja. Likovi mogu da prođu iza objekata i kreću se u tri dimenzije, a njihovi pokreti su krajnje brzi. Grad ograđen zidom može da se posmatra iz četiri ugla, koji na ekranu mogu trenutno da se prebacuju.

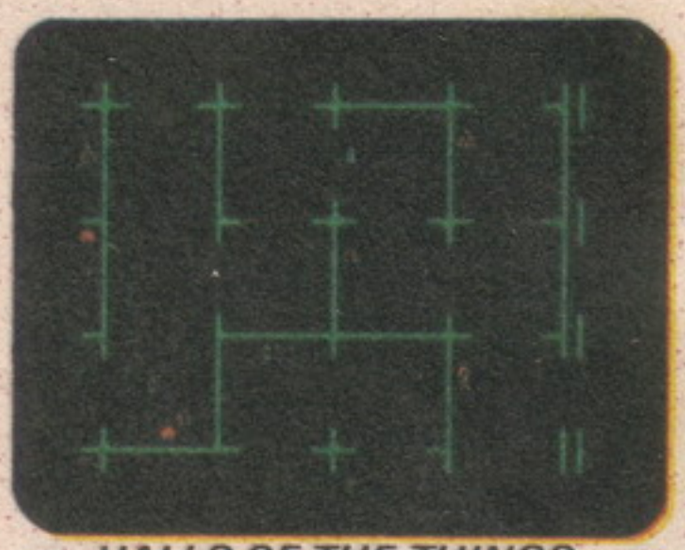
Igra je novost i po tome što vam dopušta da izaberete pol protagoniste koji se protiv zlih mrava bori za spas zatvorenika iz grada.

10 PSYTRON
Beyond

Psitron

Neobična kombinacija upravljanja zemljom i akcije uz prekrasnu grafiku čine „Psitron" jednom od najfinijih igara iz 1982. godine.

Vi kontrolišete inteligentni kompjuterski sistem koji svemirsku koloniju brani od napada. Akcija se zbiva istovremeno na deset ekrana i na većem broju nivoa. Dok napredujete kroz igru, sve više postaje neophodno planiranje, pošto morate da čuvate broj članova posade. Morate i da organizujete snabdevanje do vitalnih oblasti između bombardovanja koje vrše osvajači. Igra, takođe, pokazuje prozor za statusne izveštaje i neke od najlepših slika pozadine među akcionim igrama i igrama avanture.

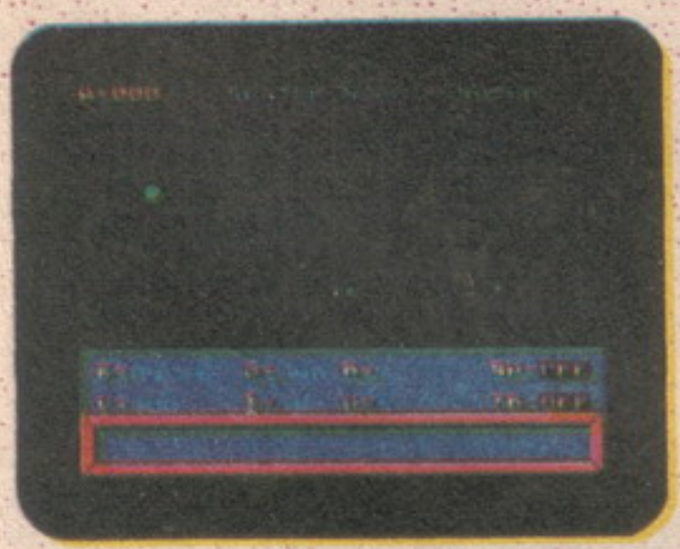


11 HALLS OF THE THINGS
Crystal Computing

Hodnici stvari

Ovaj smešni naslov otkriva jednu od prvih i najbržih akcionih igara, koja je dovela do mnogih sličnih, ali nijedna nije dostigla njeno tehničko savršenstvo.

Vi se krećete kroz lavirint sa više nivoa, napadajući čudovišta pomoću strela i munja. Stvari mogu da vas napadnu u grupi i sprže vas na smrt. Da biste pobegli iz lavirinta, morate da prikupite ključeve i zadobijete snagu iz mleka u bocama na koje uz put nailazite. Jedinstvena rutina učitavanja čini program gotovo nepristupačnim za pirate.

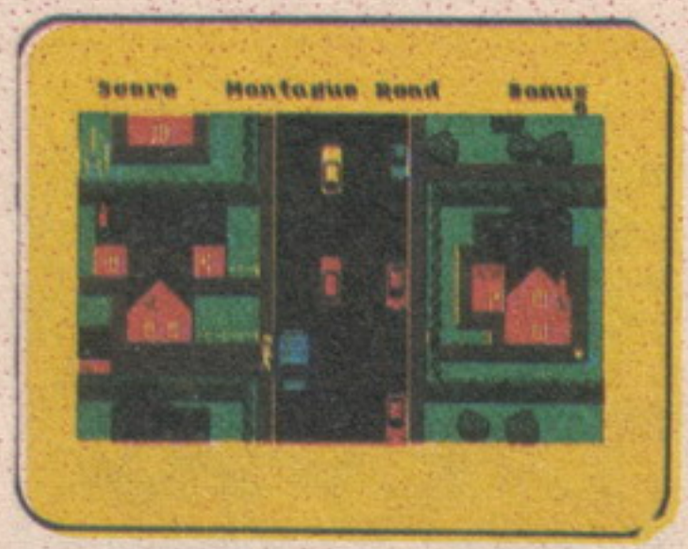


12 CODE NAME: MAT
Micromega

Šifrovani naziv: Mat

Poznata pre svega kao igra za velike kompjutere, „Šifrovani naziv: Mat" je svemirska igra avantura i strategije, koja bi mogla da se opiše kao pojačana verzija „zvezdanih staza".

Vi stupate u flotu borbenih krstarica razasutih po galaksiji kao kadet ili zapovednik. Zadatak je da odbijete Mione, koji razaraju sve planete u našem Sunčevom sistemu. Morate ih zaustaviti preko daljinskih komandi na brodovima u drugim sistemima i neposrednom borbom koja se započinje prolaskom kroz zvezdane kapije u jedan drugi deo galaksije. Trodimenzionalni efekti koji prikazuju kretanje svemirske krstarice ostvaruju se pomoću samo 2,5 kilobajta mašinca.



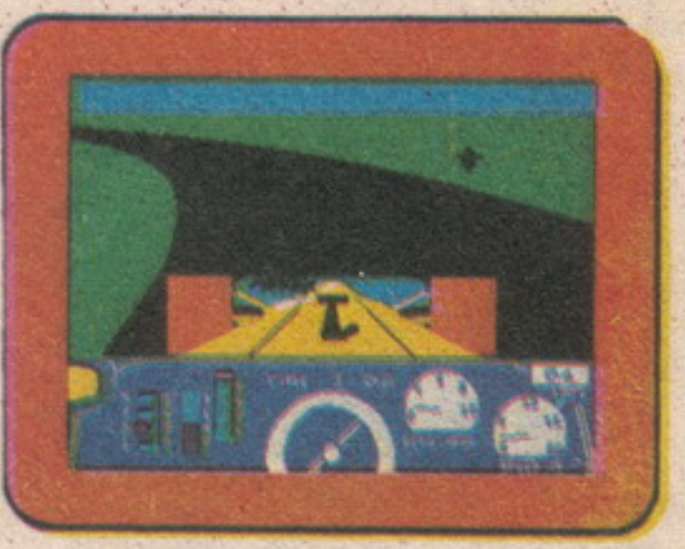
13 TRASHMAN
New Generation

Đubretar

Možda igra zasnovana na smeću ne izgleda najbolje za dobru prodaju, ali „Đubretar" je kad se pojavi 1984. godine, veoma brzo postao klasična igra.

Vi ste đubretar koji prikuplja bačene konzerve i prazni ih u svoja teretna kolica. Za svaku ulicu postoji vremensko ograničenje, a rizik uključuje automobile, bicikliste sa pločnika i opasne pse.

Ova igra sa izvanrednom grafikom, za koju je dobila francusku nagradu „Nova generacija", namenjena je čitavoj porodici. Otuda jedna „domaća" tema, u kojoj nema nasilja, i malo šale, u kojoj posmatrači mogu da uživaju. Po svom scenariju, „Đubretar" je jedna od najboljih igara. Malo je, naime, igara koje su jednako zabavne i za posmatranje i za učestvovanje.

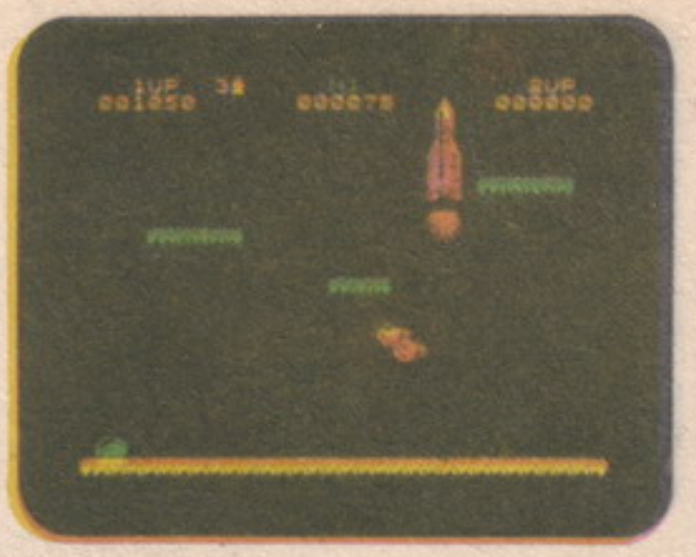


14 CHEQUERED FLAG
Psion

Zastava na kocke

U toku prve godine kada je proizvedena, ova igra je retko silazila sa rang-lista „deset najboljih". Simulacija „formule I" pružala je mnoštvo problema u programiranja, koje je firma „Psion" rešila u uobičajenom stilu i sa solidnim kvalitetom.

Igra uključuje vožnju jednih od troje izabranih kola kružnom stazom za „Gran pri", pri čemu se vodi računa o vremenu. Osim đavolski složenih promena komandi potrebnih za savladavanje staze velikom brzinom, dodatni rizik predstavljaju lokvice ulja i vode. Snaga igre je u sposobnosti simuliranja uzbuđenja o kojima mnogi od nas sanjaju, a nikada ih neće neposredno iskusiti.



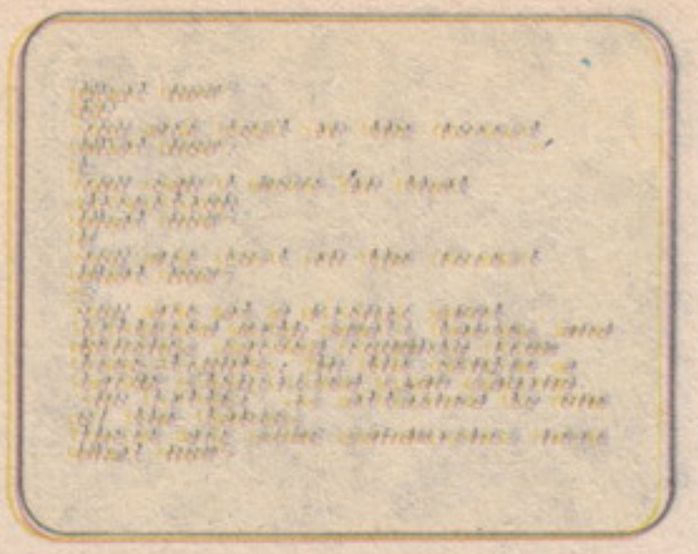
15 JETPAC
Ultimate

Mlazni ranac

Ovo je prva od dugoživotnih akcionih igara kompanije koja je svoj ugled stekla zahvaljujući kvalitetu.

Igra se zasniva na svemircu sa mlaznim rancem, koji treba da izgradi brod od raznovrsnih komadića raketa razasutih preko ekrana, a zatim pohvata rezervoare sa gorivom kako bi mogao da uzleti do sledećeg ekrana.

Jedna od najbolje prodavanih igara, „Mlazni ranac" je postavio standard za igre sa sprojtovima, pošto je čitav ekran prekriven obojenim predmetima u pokretu, na koje ne utiče uobičajena sporost računara na unošenje ili kretanje znakova.

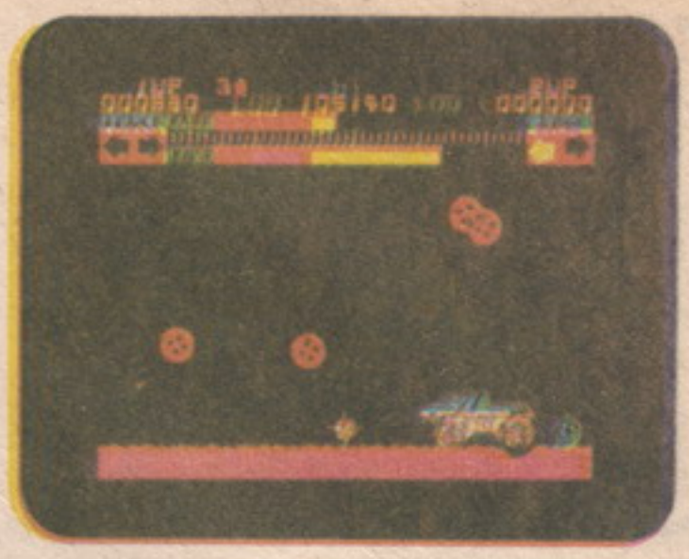


16 COLOSSAL ADVENTURE
Level 9

Kolosalna avantura

Dva gospodina, Krauter (Crowther) i Vuds (WOODS), stvorila su poodavno jednu igru avanture na glomaznoj mašini DEC i tako otpočeli čitavu tu kompjutersku scenu avantura i sličnih igara. Njihova avantura je još uvek jedna od najlepših dosad napisanih, a za „spektrum" je urađeno nekoliko verzija. Jedna od njih je i „Kolosalna avantura", sa dodatnim bonusom od 70 novih lokacija.

Koristeći napredne postupke sažimanja reči, program sadrži duge, zapetljane tekstualne opise, veliki leksikon i stotine soba i zagonetki. Program ima sva obeležja klasične verzije: tamnicu, zaplet i puno humora — i pruža stvarno osećanje zadovoljstva.



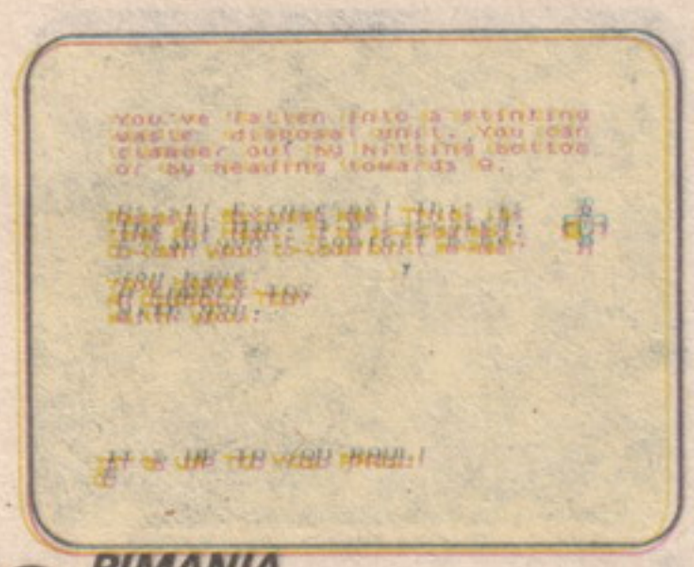
17 LUNAR JETMAN
Ultimate

Lunarni mlazni čovek

Nastavak jednako uspešnog „Mlaznog ranca", ova igra se ispoljila kao vrhunski domet u programiranju grafike.

Uz svoj mlazni ranac, svemirac u „Lunarnom mlaznom čoveku" ima na raspolaganju mesečev auto da bi istraživao negostoljubivu okolinu, otkrio položaj baze vanzemaljaca i uništio je. U međuvremenu se stvara nova baza, pa i ona mora da bude uništena.

Obeležje ove igre je gotovo potpuno odsustvo naredbi. Primorani ste da otkrijete značaj razbacanih objekata i zatim se uspešno krećete ograničeni jedino gorivom i vremenom. To i onako impresivnu akcionu igru pretvara u izazovan promišljeni poduhvat.

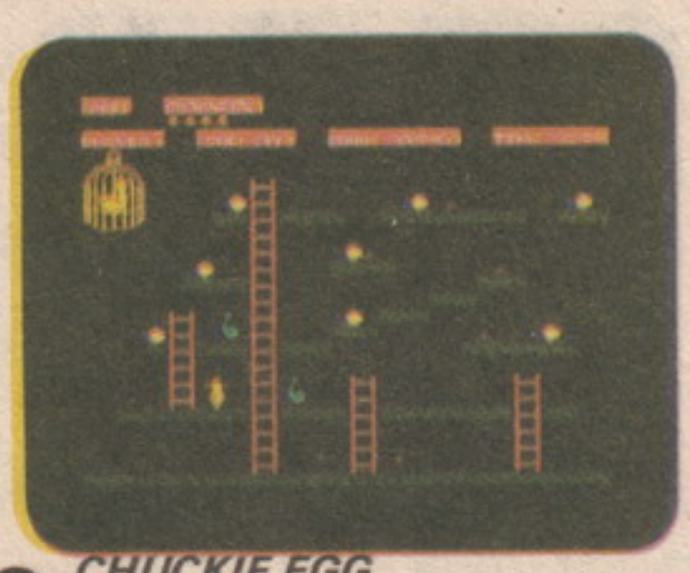


18 PIMANIA
Automata

Pimanija

Prvo pomeranje prema grafičkoj avanturi vodi vas na putovanje kroz misterioznu Zemlju Pi, u kojoj treba da pronađete i prikupite objekte pravilnim redosledom da biste orkrili Kapiju Pi i Zlatni sunčev časovnik.

Igra kombinuje zagonetke sa urnebesnom komedijom Pimanca koji se neočekivano pojavljuje s vremena na vreme i čini neverovatne stvari sa saksofonom. „Pimanija" je jedna od prvih igara u kojoj je skriven Zlatni časovnik, tajna koja vodi do nagrade. Kombinacija inovativnog softvera i izazovnog takmičenja učinila je ovu igru legendom.

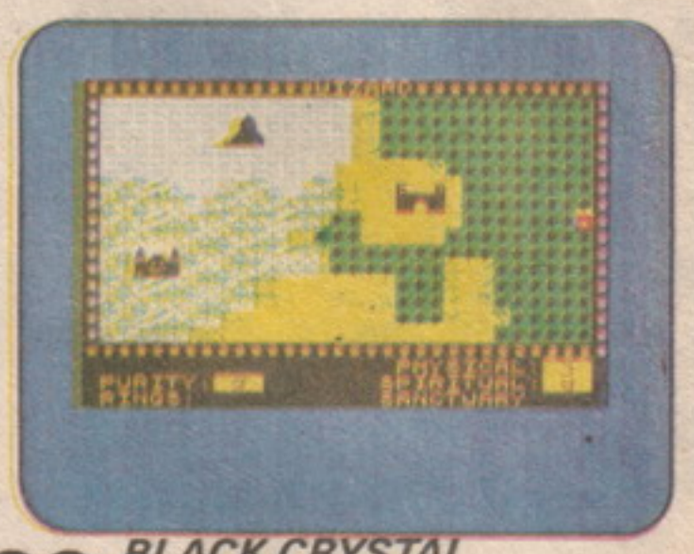


19 CHUCKIE EGG
A & F Software

Jaje miljenik

Jedna od prvih igara u više nivoa koja nas od mentaliteta svemirskih osvajača vodi do sveže i neobične radnje, „Jaje miljenik" je igra u kojoj treba da pokupite jaja dok odbijate kvočke. Kada se sva jaja prikupe, penjete se do sledećeg sprata pilićnjaka i, ako ste dovoljno srećni, stičete pozamašnu premiju.

Prvobitno napisan za 2X-81, program je potomak igara poput „Rudara manijaka". On obnavlja veru igrača u industriju softvera i podstiče stvaralačko razmišljanje u projektovanju igara.

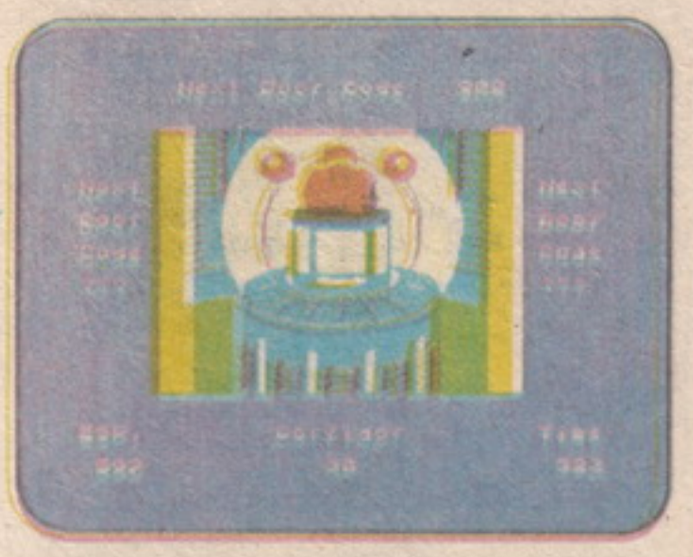


20 BLACK CRYSTAL
Carnell/Mastervision

Crni kristal

Ova igra avanture sa ulogama sastoji se od šest programa. Zadatak je da se prikupi sedam prstenova stvaranja i neutrališe đavolska moć Crnog kristala.

Autori Roj Karnel (Roy Carnell) i Stjuart Golovej (Stuart Galloway) smestili su igru u svoj sopstveni svet fantazija, Treći kontinent. Pažnja kojom su to obavili tokom nekoliko godina dala je lep rezultat. Nijedan od dvojice autora nije se nikada bavio igrama sa ulogama ili kompjuterskim igrama avantura pre nego što su počeli programiranje. Njihov rezultat je igra puna doskočica, delikatna, potpuno različita od drugih igara avanture. Njena dužina je epska, traganje izazovno, a radnja dramatična.



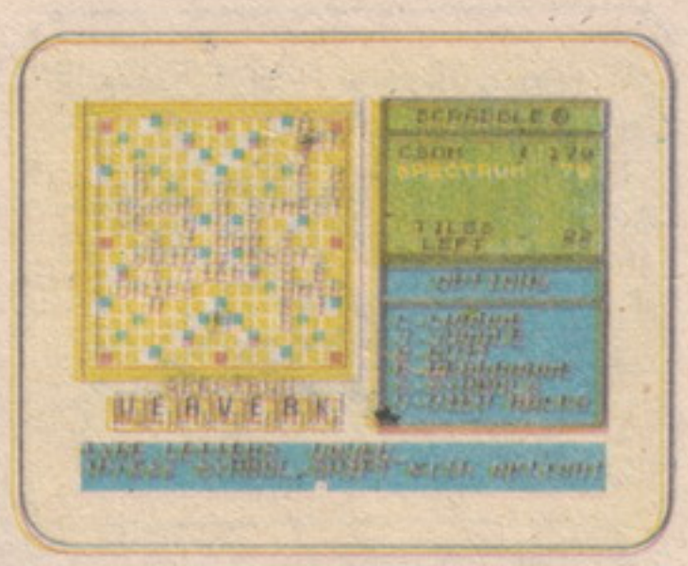
21 CORRIDORS OF GENON
New Generation

Hodnici Genona

Trodimenzionalna akciona lavirintska igra sa jednom novinom: lavirint je kružnog oblika. Vi treba da prođete kroz niz vrata od kojih neka mogu da se otvore samo uz pomoć tačne šifre-broja. Kad jednom prođete sva vrata, ulazite u sobu sa glavnim kompjuterom da biste onesposobili mašinu, što će uništiti čitav kompleks.

Jedini problem je Bogul-stvorenje koje se čudno oglašava i crpi psi-energiju iz svakog igrača kojem se približi. Bogul ume samog sebe da klonuje, pa ćete na povratku morati da izbegnete mnoge takve iste stvorove.

Igra spaja najbolje elemente strategije i akcije.



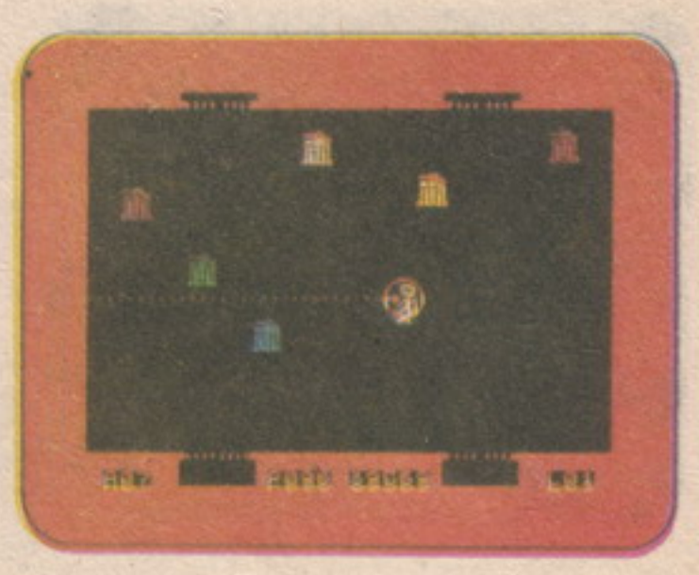
22 SCRABBLE
Psion

Slagalica reči

Ovo je jedina verzija popularne igre na tabli koja je adaptirana za „spektrum" i to je najbolja verzija urađena za bilo koji mikrokompjuter.

Ovu slagalicu reči mogu da igraju četiri igrača. Ako igra samo jedan, kompjuter može da zameni preostalu trojicu, na nivoima koje njegov ljudski sutakmičar izabere. Program sadrži rečnik od 11000 reči i odgovara za 90 sekundi. Moguće su i podvale, tako što se kompjuter navede da poveruje u tačnost reči, ali to nije sportski.

„Slagalica reči" je najbolja simulacija igre na tabli i sigurno prilično inteligentna. Primenjene tehnike „pakovanja" reči ostavljaju prostor u memoriji za sjajnu grafiku i kompjuterovu sposobnost da igra za trojicu.

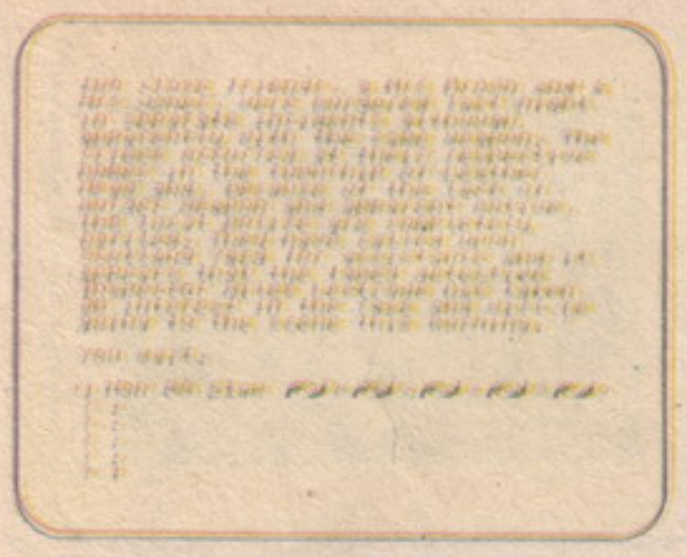


23 THE PYRAMID
Fantasy

Piramida

U ovoj akcionoj igri avanture u kompjuterski svet se uvodi Zigi, vanzemaljac koji namerava da otkrije tajnu života, vasione i svega drugog.

On u svom traganju ispituje piramidu koja sadrži mnoga vanzemaljska stvorenja koja treba izbeći ili uništiti. Zigi mora da prođe kroz niz komora i nivoa, počinjući u vrhu i završavajući na donjem nivou sa konačnim izlazom. Postoji 120 komora i 15 nivoa, tako da je uspešnom igraču potrebno nekoliko meseci da se usavrši. Svi igrači mogu da se uključe u takmičenje sa visokim zbirom poena i imaju izglede da budu pohvaljeni u biletnu zemlje Fantazije.

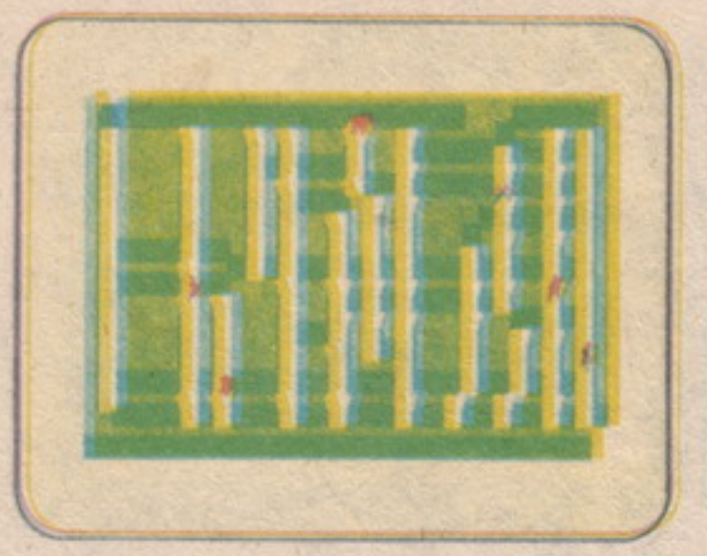


24 SHERLOCK
Melbourne House

Šerlok

Ovaj naslednik klasične igre broj jedan, „Hobita", napravljen je za godinu i po dana razvoja — šest meseci više nego što je bilo planirano. Rađen u velikoj meri na koncepciji interakcije likova primenjivanoj kod ranih igara, „Šerlok" uvodi veliko mnoštvo likova i povećanu usavršenost komandi.

Igra od vas zahteva da odgonetnete tipičnu zamršenu misteriju upotpunjenu sa mnogo sporednih detalja, koji stvaraju autentičnu atmosferu — iz priča o Šerloku Holmsu. Uprkos slabašnoj grafici, igra je veoma zanimljiva i zauzela je trajno mesto u srcima ljubitelja avanturističkih igara.

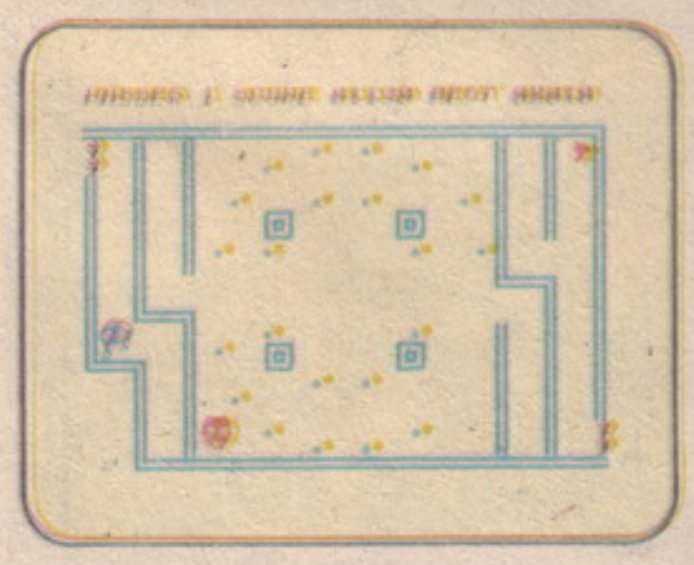


25 ESCAPE
New Generation

Bekstvo

Ovo je prva među trodimenzionalnim igrama. Igrač tumara kroz lavirint pokušavajući da pokupi jednu sekiru sa namerom da pobegne od nekoliko (do pet) dinosaura koji ga progone. U petom nivou igre, koji je najteži, oživljeni veliki pterodaktil uznemirujuće mlati krilima duž staza lavirinta u potrazi za plenom.

„Bekstvo" je jedna od prvih igara za „spektrum" ali i posle dve godine zadržala je svoj tehnički sjaj. Ona je legenda među vlasnicima ovog računara, čije sećanje seže koju godinu unatrag, i još uvek spada među najomiljenije.

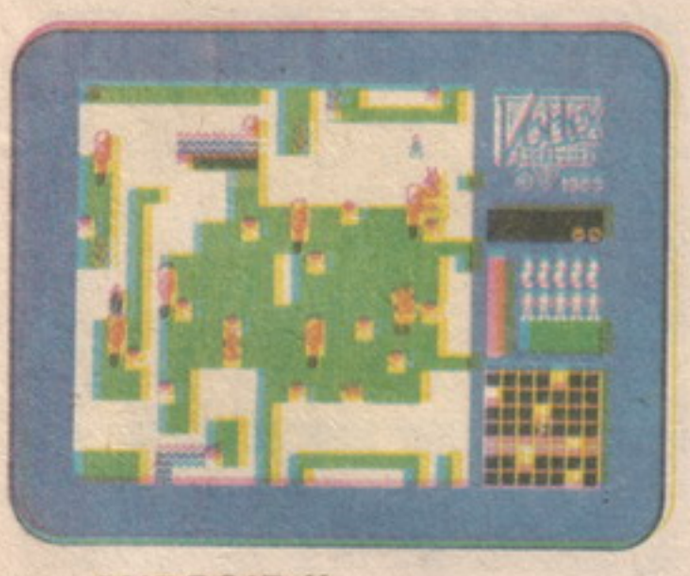


26 HUNGRY HORACE
Melbourne House

Gladni Horacije

Glavna ličnost ove popularne igre je Horacije, lik rađen po ugledu na strip, koji klopa biljke u parku-lavirintu i beži od čuvara koji nastoje da ga uhvate. Ovo je prva igra u kojoj je junak važniji od radnje. Grafika je bila i ostala izvanredna, sa dominantnom upotrebom sprojtova za oslikavanje likova, tehnikom koja ranije nije bila korišćena.

Igra je logičan deo razvoja poteklog od „Pakmena", koji, inače, nema softverski ekvivalent za „spektrum", pošto firma „Atari" (vlasnik „Pakmena") nema interesa za male kompjutere.



27 ANDROID II
Vortex

Android II

Impresivna trodimenzionalna akciona igra, „Android II", mnogo se više oslanja na izgled nego na radnju. Robot-junačina mora da zaustavi napredovanje Militoida, rase stonoga, i uz put odbije robote čuvare Lebdodopide i Izbacivače, pri čemu moraju da zaobilaze mine na tlu. U igri postoje tri zone — lavirint smrti, zona paradoksa i ravne zemlje. Nije verovatno da će igrači koji dosegnu drugu fazu igre ostaviti na miru treću.

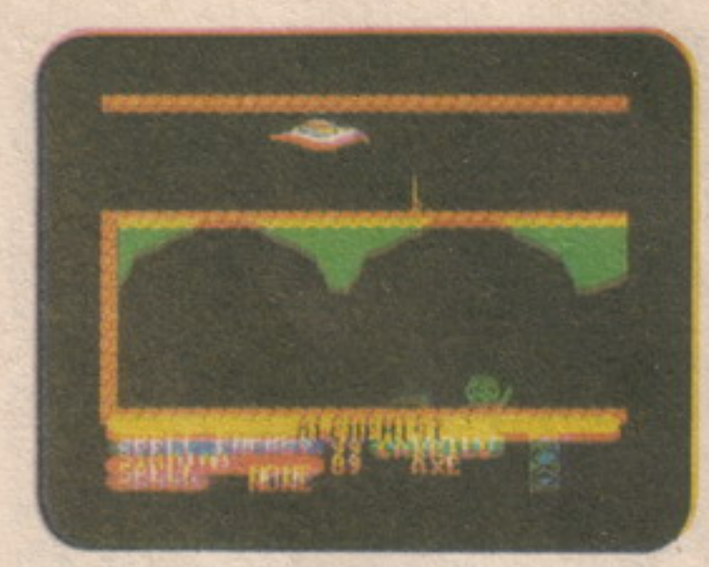
Razvijena krajem 1983, igra je bila jedna od prvih koja generiše nasumični lavirint veći od ekrana i koja se skroluje u svim pravcima.

28 FULL THROTTLE
Micromega

Puni gas

Simulacija automobilskih trka, koja se kretala putem koji je utrla „Zastava na kocke", postpeno je gubila dah. Firme Micromega se tada odlučila radije za „Gran pri" trku za motorcikle, čija je odlična verzija „Puni gas".

U osnovi slična prethodnoj igri, ova nova ima jedno bitno poboljšanje — stvarnu trku, u kojoj učestvuje četrdeset drugih motorciklista. Ostalo je slično: zavojita staza i retka pozadina. Tamo gde grafika nije nemirna, igra može da se ocenjuje još detaljnije.



29 THE ALCHEMIST
Imagine/Beau-Jolly

Alhemičar

Spoj akcionih veština sa logičkim razmišljanjem obezbedio je igrama avanture ogromnu popularnost još od „Hodnika stvari". „Alhemičar", pravi praznik za oči, izazovna je igra, jedna od najboljih.

Potrebno je da brzo kombinujete sastojke za smesu čaroliju kojom ćete osujetiti zlog čarobnjaka. Da biste to postigli, morate da se krećete kroz kompleks soba i pećina, od kojih svaka zahteva drugačiju strategiju za izbegavanje rizika. Možete da hodate u ljudskom obličju ili da se preobrazite u veličanstvenog orla i letite između nivoa ili preko ponora. Postoje mnogi objekti i male čini koje treba otkriti i koristiti u odgovarajućim prilikama.

30 THE BIRDS AND THE BEES
Bug-Byte

Ptice i pčele

U ovoj igri sa grafikom koja podseća na strip, pčela Boris nastoji da prikupi nektar. Pri tom nailazi na suprotstavljanje lokalnih ptica koje nisu zadovoljne što im neko upada u vazdušni prostor. Igra se pojavila u vreme kada je firma Melbourne House učvrstila popularnost svog junaka Horacija i kada su u modu došle oštroumne igre.

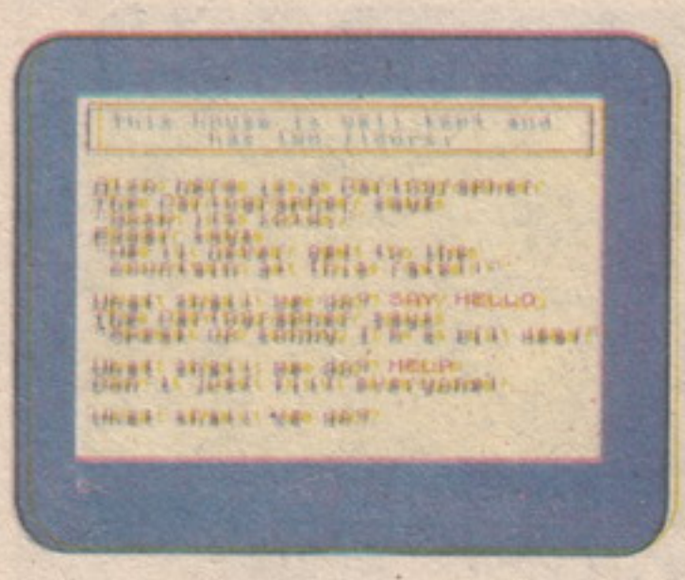
„Ptice i pčele" su trenutno postigle uspeh zato što su se razlikovale od svega što je firma Bug-Byte dotad pružila. Takođe, ova igra je pokazala da jednostavna koncepcija radnje može da se preokrene u originalnu i veoma zanimljivu igru.

31 SABRE WULF
Ultimate

Sabljasti vulk"

Naslednik pre nego neposredni nastavak „Atik ataka", „Sabljasti vulk" (u originalu „wulf", očigledno od reči „wolf" — vuk) je džinovska lavirintska igra smeštena u čudnu džunglu punu razuzdanog života, uglavnom negostoljubivog. Vi treba da sakupite četiri dela amuleta za bekstvo, pri čemu morate da se suprotstavite raznoraznim zverima. Magične orhideje naglo se rascvetavaju na vašem putu, a mnoštvo predmeta na koje nailazite može da vam pomogne u napredovanju.

Grafika je krajnje bogata i promenljiva, mada ne tako dobro uređena kao ranije igre ove firme. Područje ove lavirintske igre je ogromno i ona se bitno razlikuje od drugih igara.

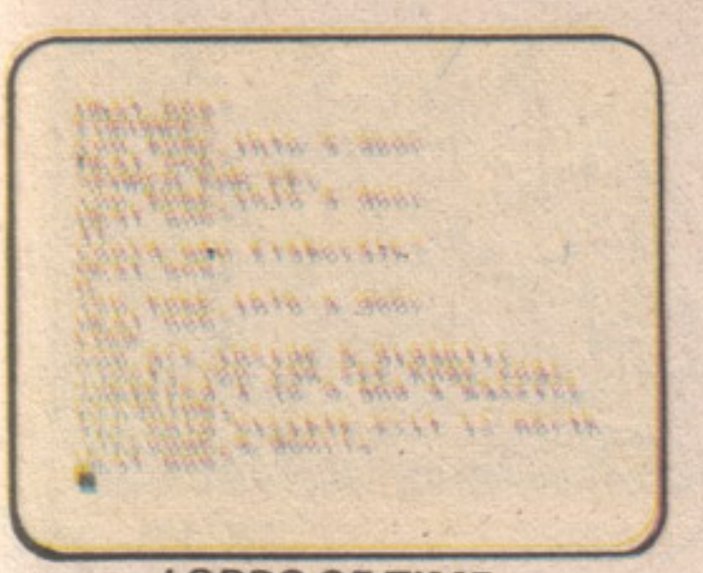


32 MOUNTAINS OF KET
Incentive

Planine Keta

Prva u trilogiji isključivo tekstualnih igara avanture stavlja igrača u opasne situacije, pri čemu mora da se bori protiv raznih likova kojima je cilj da nanesu što veću štetu.

Igra sadrži novi sistem borbe koji je dosta slobodno nadograđen na metod bodovanja kod „Tamnica i zmajeva", gde se lik određuje na osnovu sistema tačaka. „Planine Keta" su neverovatno detaljne u pogledu radnje i atmosfere i sadrže čak i monetarni sistem. Autor je uključio interaktivne likove koji se, mada ne naročito inteligentni, ravanju po vlastitoj volji u odnosu na odluke koje donosi igrač.



33 LORDS OF TIME
Level 9

Gospodari vremena

Ova avantura samo sa tekstom zahteva od vas da proputujete različite vremenske zone na dedinom časovniku, prikupljajući predmete kojima ćete moći da odbijete Gospodare vremena. Igra uključuje preko 200 lokacija i 80 objekata. Svedena je na 48 K zahvaljujući specijalnom kompajleru za sažimanje podataka.

Igra spada u svoju sopstvenu klasu, jer sadrži avanture u avanturi. Postoji ukupno devet i svaka ima barem tri scene. složenost radnje može da okupira igrače mesecima i daje igri stalno mesto u istoriji igara avanture.

34 URBAN UPSTART
Richard Shepherd Software

Gradski skorojević

Ovo je istinska avantura za osamdesete godine, smeštena u grad Skartorp, ćorsokak u turobnom svetu unutar-gradskog raspada. Vaš jedini zadatak je da preživite.

Ovde scenario, a ne izuzetnost programiranja ili grafike, kvalifikuje igru kao klasičnu. Skartorp je nadasve obeshrabrujuće mesto. Park, prividni raj mira i zelenila, omeđen je farmom otpadaka i grobljem. Grupe policajaca i fudbalskih huligana tumaraju ponoćnim ulicama. Ova firma zaslužuje priznanje za nov i potpuno različit pristup tržištu avantura.

35 FOOTBALL MANAGER
Addictive Software

Fudbalski menadžer

Napisana u bejziku i ne najbrža igra ikad viđena, „Fudbalski menadžer" je ipak postigao veliki uspeh i na „spektrumu" i na „ZX-81".

Vaš posao je da provedete fudbalski tim kroz sezonu u ligi i kroz takmičenja u kupu. Svakom igraču se stalno ocenjuju „rejting" veština i energija, a vaš tim dobija ocene za moral u zavisnosti od njegovog stanja u poslednjih nekoliko igara. Igra se izvodi u nizu završnica pred golom, dok vi sedite i grickate nokte. Obmanjujuće jednostavna, ovo je po svoj prilici najkorišćenija igra strategije ikada proizvedena za „spektrum".

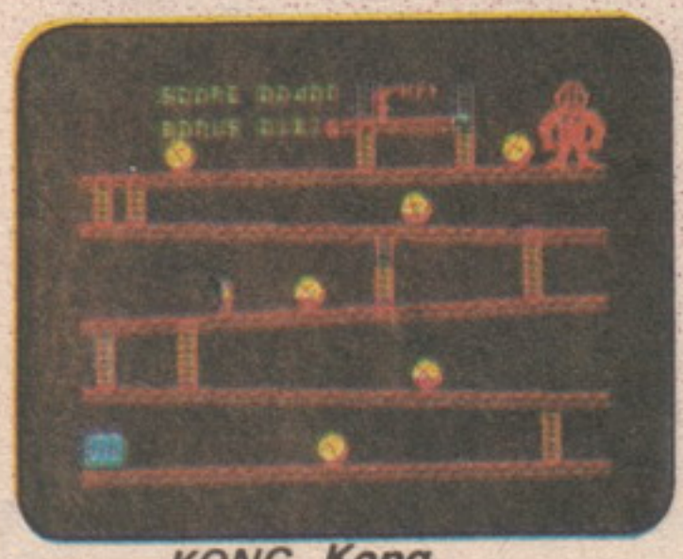
36 TLL
Vortex

TLL

Vrhunsko delo u programiranju trodimenzionalne grafike, ovo je igra u kojoj upravljate lovcem „tornado" sa mašućim krilima nekoliko metara iznad tla da biste eliminisali niz ciljeva.

Grafika su solidno nacrtane kuće, stabla i telegrafske linije. „Tornado" baca senku, koja se čak vere uz zidove zgrada. Možete da upravljate ispod mostova i telefonskih linija bez ikakvog treperenja na ekranu.

Iako nije najteže među igrama, prostrano područje za igranje i krasna grafika daju ovoj igri prednost nad mnogim takmacima i čine je pravom uživancijom.



37 KONG
Ocean

Kong

Ukusi u pogledu akcionih igara su različiti, i dok kritičari nastoje da ocene originalnost i usavršenost, publika često više drži do svojih starih ljubimaca iz sveta akcije. „Kong" je jedna takva igra, direktno preuzeta od tvoraca akcionog hita „Magarac Kong".

Kao Mario, vi morate da spasete prijateljicu iz šapa džinovskog gorile, penjući se preko mnoštva nosača i druge „projektile". Bogata grafika i prava mešavina veštine i brzine čine ovu igru jednim od najuspešnijih pokušaja u prevođenju igre sa automata na televizijski ekran.

38 VELNOR'S LAIR
Quicksilva

Velnorova jazbina

Duhovita, zamršena tekstiilna avantura poput „Tamnice i zmajeva", „Velnorova jazbina" vas vodi u tamnicu-tvrđavu zbog čarobnjaka Velnora, da biste sprečili da on uništi svet. Igra je značajna po svom jednostavnom ali efikasnom sistemu borbe i po izboru likova. Igrači mogu da startuju u jednoj od tri klase. Borci su snažni, čarobnjak koristi čini, a sveštenici se sami leče.

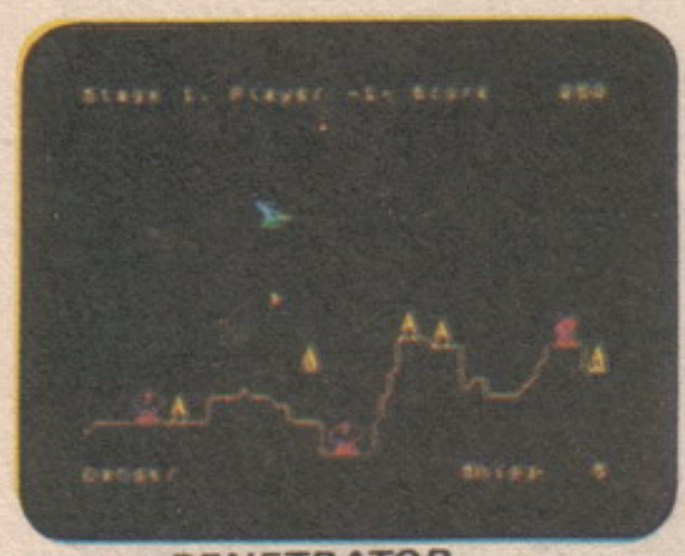
Malo je igara avanture koje uspevaju da spoje logičke probleme i borbu protiv čudovišta i „Velnorova jazbina" ostaje i jedna od najboljih i jedna od najranijih. Dobro urađena i dosta potcenjena igra.

39 ROMMEL'S REVENGE
Crystal Computing

Romelova osveta

Ovo je najbolja verzija akcione igre „Zona bitke" na „spektrumu", u kojoj vaš tenk treba da uništi neprijateljski. Igra se razlikuje od originala za „atari" i sadrži izvesna poboljšanja. Možete da uključite ili isključite vatrene eksplozije iz vulkana na brdu. Programske opcije, takođe, dopuštaju slabašne zvučne signale koji nagoveštavaju da je neprijateljski tenk u blizini.

Puštena u prodaju krajem 1983, ova igra ima izvanrednu grafiku i jedna je od malog broja igara u kojima se trodimenzionalna perspektiva ne uništava naglim pokretima objekata na ekranu.



40 PENETRATOR
Melbourne House

Penetrator

Jedna od najuspešnijih zabavnih akcionih igara bila je „Poletanje", u kojoj vi pilotirate letelicom kroz pećine i iznad planina do velikog skladišta za bombe.

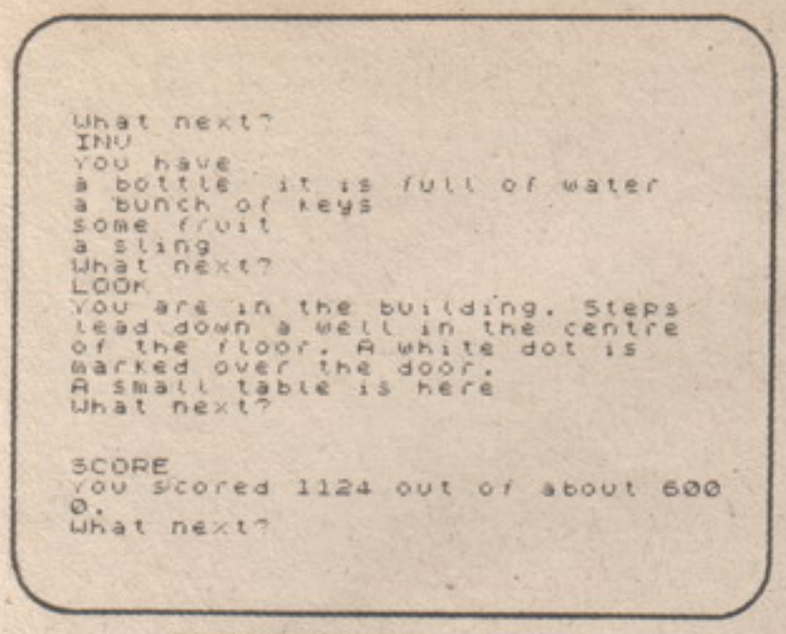
„Penetrator" je izvanredna varijanta ove igre. Oduševio je kritičare početkom 1983. godine. Grafika je prostrana i ravnomerna, sa oštrim konturima, a akcija je veoma brza i primamljiva. Ono što doista uzdiže ovu igru iznad njenih parnjaka je revolucionarna sposobnost potpunog menjanja pejzaža. Zajedno sa praktičnim modalitetima na svakom nivou, to je čini jednom od najkompletnijih akcionih igara do danas.

41 MUGSY
Melbourne House

Magsi

Posredi je širenje teme upravljanja zemljom, pri čemu umesto vladanja malim kraljevstvom igrač preuzima ulogu šefa podzemlja koji nastoji da se nezakonito naguma što veće količine love pre nego što ga spreče. Novac za podmićivanje se redovno isplaćuje, a policija često navraća do razbojnikovog ličnog sefa da bi pokupila pare za svoj dobrotvorni fond.

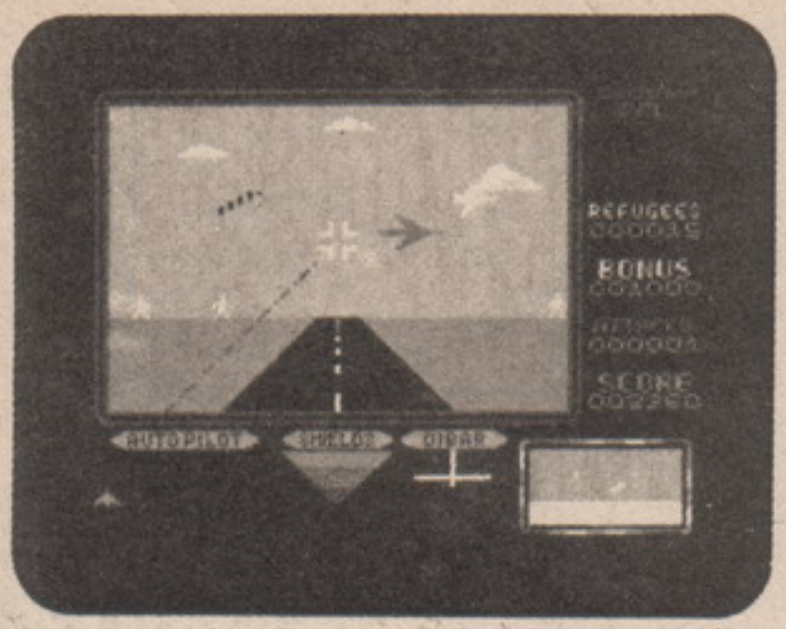
„Magsi" se razvija kao niz slika u animiranom filmu. Naredbe i odgovori smeštaju se u oblačiće za govor, što igri obezbeđuje novi kvalitet. Koncepciju stripa kritičari su dobro primili, kao i kupci.



42 ADVENTURE QUEST
Level 9 — **Potraga za avanturom**

Nastavak „Kolojalne avanture, originalne igre avanture za velike kompjutere, i druga u trilogiji „srednja zemlja" ove firme, igra „Potraga za avanturom" prepoznaje se po svojim poetskim tekstualnim opisima i možda se približava literaturi više nego ijedna tekstualna igra avanture.

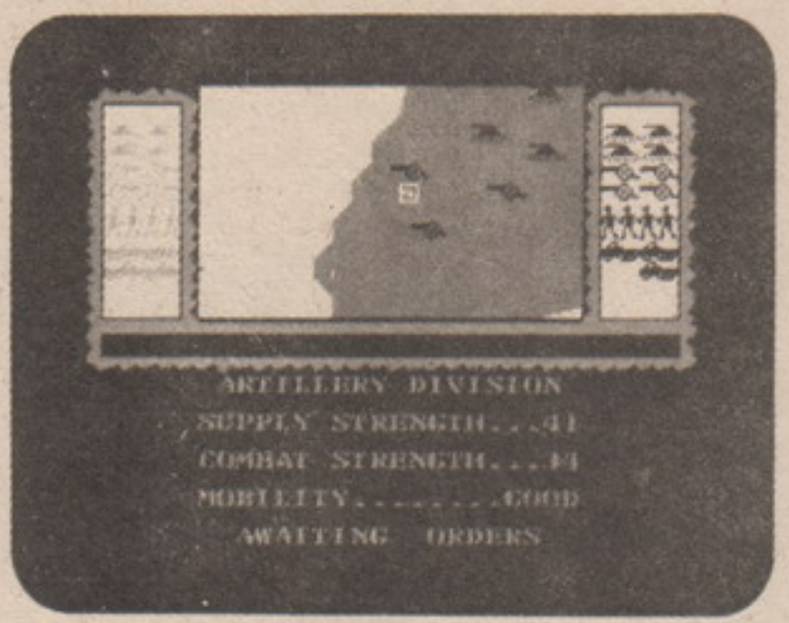
Premda je najveći deo akcije smešten u pustinje, planine, šume i uz misteriozne reke, igra neprekidno pravi vezu sa svojim prethodnikom u trilogiji, i po radnji i po upotrebljenom rečniku. Mada nema grafike, Nivo 9 je jedna od malobrojnih kompanija koja jezik koristi da bi u našoj mašti stvarala slike.



43 ZZOOM
Imagine/Beau-Jolly — **Zum**

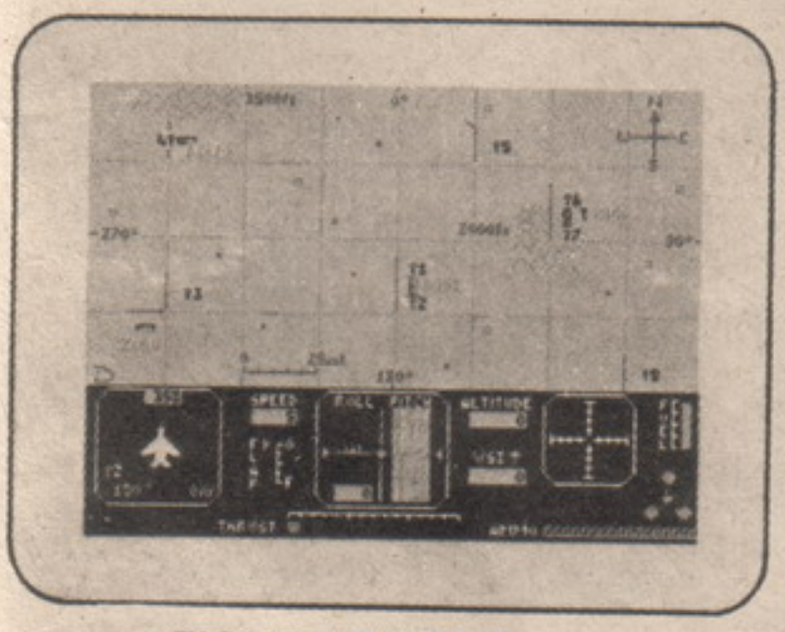
Reč je o akcionoj igri u kojoj treba da odbranite lanac skloništa od neprijateljskih aviona, tenkova i podmornica. Grafika je ovu igru svrstala među odabrane čim je u jesen 1983. godine puštena u prodaju. Svi likovi i detalji kreću se preko ekrana ravnomerno, na izgled sasvim nezavisno od drugih. Nema nikakvih teškoća sa preklapanjem boja.

Igra je veoma primamljiva. Izgleda da likovna obrada, koja je rođena po jednom modernom scenariju, mnogo više okupira pažnju igrača nego prosečna akciona igra. Dok napredujete kroz različite nivoe, nasilje prerasta u pravu orgiju razaranja.



44 STONKERS
Imagine/Beau-Jolly — **Stonkeri**

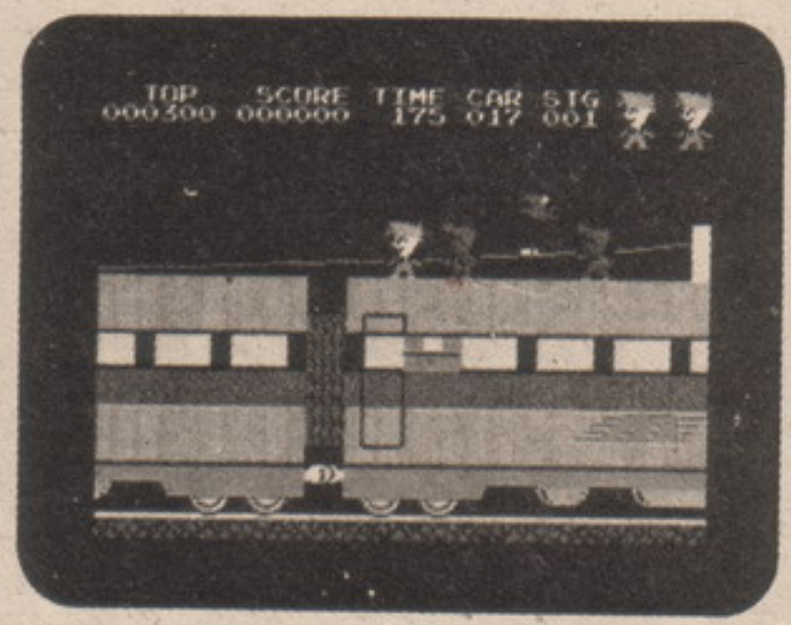
Ovo je veoma doterana ratna igra, u kojoj dovodite u red tenkove i artiljeriju da biste se tukli sa slično opremljenim neprijateljem. Igra je vizuelno privlačna jer sadrži mapu terena sa ucrtanim položajem obeju snaga. Određeni ključevi omogućuju vam da zumirate pojedine odseke mape. Naporedo sa borbom, morate se uveriti da su namirnice, istovarene iz broda u zalivu, upućene na borbene položaje. Igra takođe uključuje neke složene logične rutine koje kompjuteru omogućuju da postane težak protivnik.



45 FIGHTER PILOT
Digital Integration — **Pilot lovca**

Posredi je jedna od najboljih simulacija letenja na tržištu. U nečemu nadmašuje i Psionovu „Simulaciju letenja": dopušta vam ne samo da pilotirate modernim mlaznim lovcem nego i priređuje simuliranu vazdušnu bitku. Ateriranje lovcem F-15 mnogo je teže nego sa lakim lovcem, zbog čega je upotreba programa složena, ali predviđeni su praktični modaliteti koji razvijaju vašu veštinu.

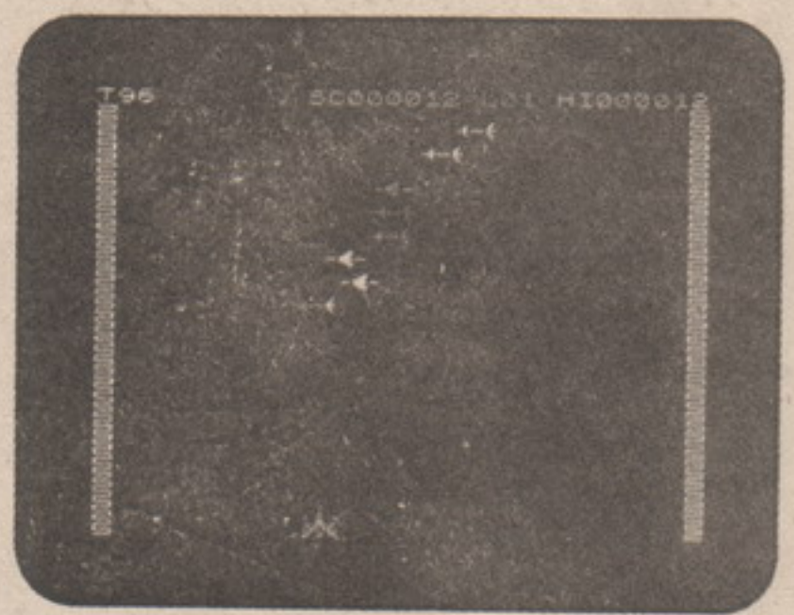
Povećana prefinjenost programa postignuta je na štetu grafike koja predstavlja pejzaž, tako da nemate priliku da se zabavljate zaranjanjem u jezero i sličnim zanimacijama.



46 STOP THE EXPRESS
Sinclair Research — **Zaustavite ekspres**

Ova akciona igra rađena je po uzoru na strip, sa neobičnom radnjom. Igrač je tajni agent. On mora da zaustavi voz pre nego što stigne do zemlje crvenog (?) čoveka, koji je oteo voz. Specijalni agent može da uspori voz time što će pretrčati preko vrha čitave kompozicije, ući u ekspres i stići do motora, koji može da se zaustavi. Uz put mora da izbegava crvenog čoveka i njegove smrtonosne pištolje.

„Zaustavi ekspres", koji se pojavio na tržištu u proleće 1984. godine, uspostavio je novi standard za grafiku na „spektrumu".

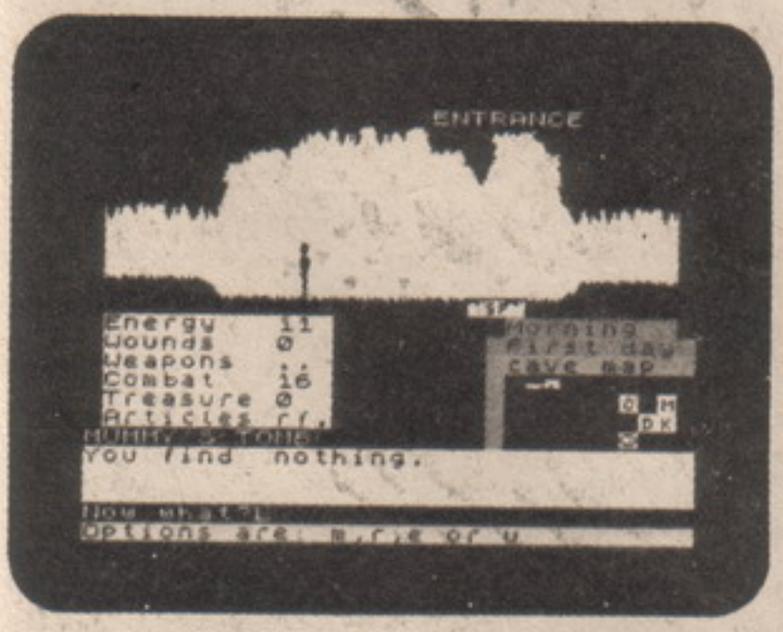


47 ARCADIA
Imagine/Beau-Jolly — **Arkadija**

Kada je ova igra banula u svet, bila je pozdravljena kao najprijemčivija igra gađanja ikad napisana. Za grafiku se govorilo da je dostigla nove visine među akcionim igrama za „spektrum".

Upoređenje igre sa važećim standardima teško može da objasni otkud sva ta larma. Grafika se danas čini grubom, a sama igra je nekakva varijanta „Svemirskih osvajača" sa dvanaest zaklona za vanzemaljce.

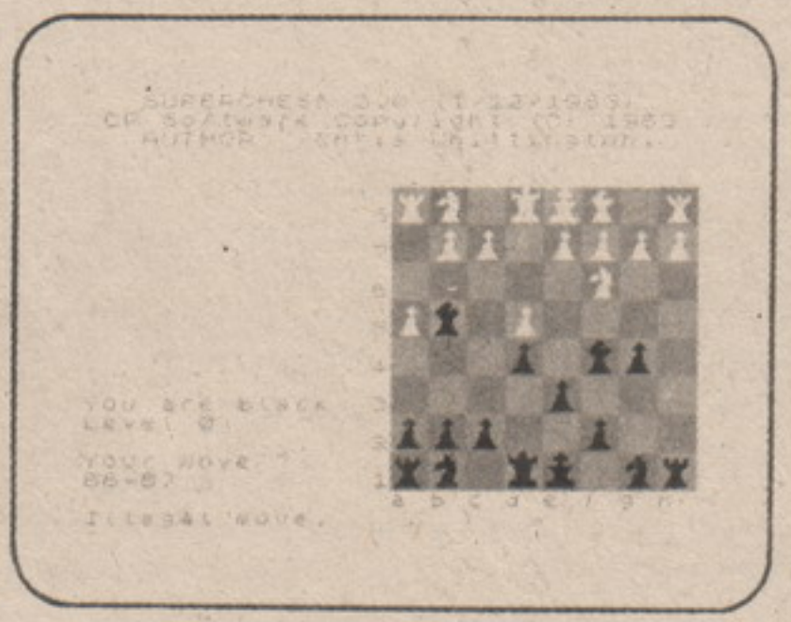
Uprkos manama, igra je vrlo prijemčiva i veoma brza. Postigla je veliki uspeh, koji je firmu Imagine doveo u vrh softverske industrije, a zatim je odveo u propast.



48 THE ORACLE'S CAVE
Doric — **Oraklova pećina**

Reč je o igri avanture sa nasumično generisanim tamnicama bez mnogo zapleta i — koherentne logike. Igra u izvesnoj meri trpi zbog tih slabosti, ali to nadoknađuje na drugoj strani.

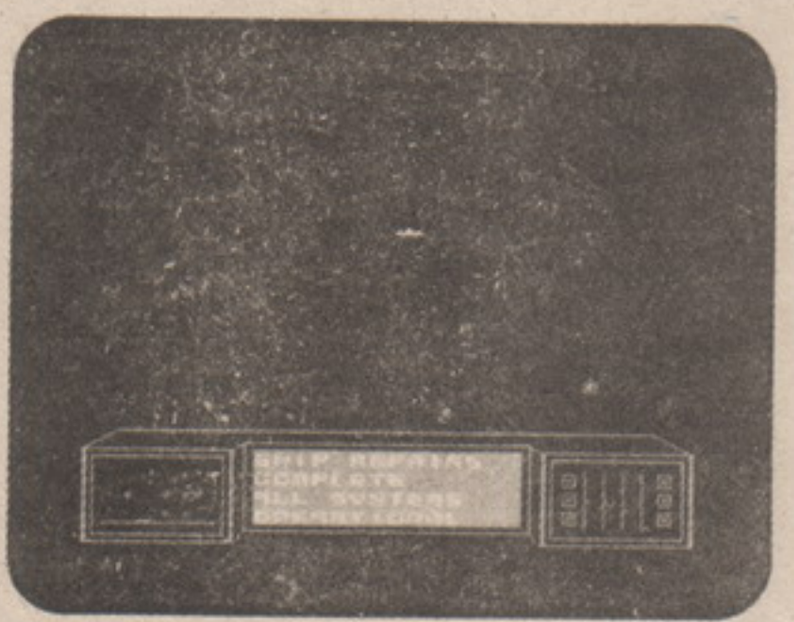
U jednoj od prvih pravih grafičkih igara avanture, vaš junak istražuje mrežu pećina, boreći se uz put sa mnoštvom čudovišta. Imate mogućnost da birate između četiri različite potrage i morate da dokrajčite igru za ograničeno vreme. Animirana grafika bila je u trenutku nastanka novost. Oskudna sa problemima, ali obilata u nasilju, „Oraklova pećina" je stekla mnoge sledbenike otkako je lansirana početkom 1983. godine.



49 SUPERCHESS III
CP Software — **Superšah III**

Posredi je igra koja omogućuje igranje na takmičarskom nivou i koja uključuje mnoge karakteristike kakve se inače ne nalaze u kompjuterskim šahovskim programima. Ima deset nivoa igre, od kojih su svi podvrgnuti turnirskim vremenskim ograničenjima i uobičajenim opcijama, kao što je preporučeni potez. Unete su i mogućnosti postavljanja i rešavanja problema. Posebne opcije uključuju informaciju o načinu na koji program „misli", displej koji pokazuje vrste poteza koje kompjuter razmatra i broj procenjenih poteza.

Igra je naslednik „Superšaha III" i koristi usavršene logičke rutine.



50 TIME GATE
Quicksilva — **Kapija vremena**

Prva trodimenzionalna akciona igra za „spektrum" proizvedena 1982. godine izazvala je različite reakcije kod kritičara. Međutim, ona je zadržala svoju popularnost kod igrača — što predstavlja iznenađenje na tržištu koje je prilagodljivo i sklono promenama ukusa i raspoloženja.

Radnja uključuje pokušaj igrača da locira bazu opakih vanzemaljaca koji nameravaju da zauzmu galaksiju. Da bi to postigao, igrač koristi vremenske staze do kojih dolazi kroz kapije vremena. Igrač treba da postepeno zaposedne sve kapije vremena i uništi sve vanzemaljce.

ćirilica na „epsonu"

Generisanje naših slova u printerima epson RX-80, FX-80 i FX-100

```
10 ! ** PROGRAM ZA PROMENU KARAKTERA KOD PRINTERA "EPSON" FX-80 I FX-100 **
20 DIM B$$100C
30 RESET 10
40 CONTROL 10,3 ; 8
50 CONTROL 10,4 ; 3
60 B$=CHR$(27)&"R"&CHR$(0)
70 OUTPUT 10 USING "#,K" ; B$
80 B$=CHR$(27)&":"&CHR$(0)&CHR$(0)&CHR$(0)
90 OUTPUT 10 USING "#,K" ; B$
100 B$=CHR$(27)&"&"&CHR$(0)&CHR$(64)&CHR$(64)
110 OUTPUT 10 USING "#,K" ; B$
120 ! ***** Ž *****
130 B$=CHR$(139)&CHR$(0)&CHR$(66)&CHR$(4)&CHR$(202)&CHR$(152)&CHR$(210)&CHR$(
32)&CHR$(66)
140 B$=B$&CHR$(0)&CHR$(0)&CHR$(0)
150 OUTPUT 10 USING "#,K" ; B$
160 B$=CHR$(27)&"&"&CHR$(0)&CHR$(96)&CHR$(96)
170 OUTPUT 10 USING "#,K" ; B$
180 ! ***** Ž *****
190 B$=CHR$(139)&CHR$(0)&CHR$(34)&CHR$(4)&CHR$(162)&CHR$(72)&CHR$(162)&CHR$(1
6)&CHR$(34)
200 B$=B$&CHR$(0)&CHR$(0)&CHR$(0)
210 OUTPUT 10 USING "#,K" ; B$
220 B$=CHR$(27)&"&"&CHR$(0)&CHR$(91)&CHR$(94)
230 OUTPUT 10 USING "#,K" ; B$
240 ! ***** Š *****
250 B$=CHR$(139)&CHR$(36)&CHR$(82)&CHR$(0)&CHR$(210)&CHR$(128)&CHR$(210)&CHR$
(0)&CHR$(82)
260 B$=B$&CHR$(12)&CHR$(0)&CHR$(0)
270 OUTPUT 10 USING "#,K" ; B$
280 ! ***** Đ *****
290 B$=CHR$(139)&CHR$(16)&CHR$(130)&CHR$(254)&CHR$(130)&CHR$(146)&CHR$(130)&C
HR$(130)&CHR$(68)
300 B$=B$&CHR$(56)&CHR$(0)&CHR$(0)
310 OUTPUT 10 USING "#,K" ; B$
320 ! ***** C *****
330 B$=CHR$(139)&CHR$(60)&CHR$(66)&CHR$(0)&CHR$(66)&CHR$(128)&CHR$(66)&CHR$(0
)&CHR$(66)
340 B$=B$&CHR$(36)&CHR$(0)&CHR$(0)
350 OUTPUT 10 USING "#,K" ; B$
360 ! ***** Č *****
370 B$=CHR$(139)&CHR$(60)&CHR$(66)&CHR$(0)&CHR$(194)&CHR$(128)&CHR$(194)&CHR$
(0)&CHR$(66)
380 B$=B$&CHR$(36)&CHR$(0)&CHR$(0)
390 OUTPUT 10 USING "#,K" ; B$
400 B$=CHR$(27)&"&"&CHR$(0)&CHR$(123)&CHR$(126)
410 OUTPUT 10 USING "#,K" ; B$
420 ! ***** š *****
430 B$=CHR$(139)&CHR$(16)&CHR$(42)&CHR$(0)&CHR$(170)&CHR$(64)&CHR$(170)&CHR$(
0)&CHR$(42)
440 B$=B$&CHR$(4)&CHR$(0)&CHR$(0)
450 OUTPUT 10 USING "#,K" ; B$
460 ! ***** đ *****
470 B$=CHR$(139)&CHR$(12)&CHR$(18)&CHR$(0)&CHR$(18)&CHR$(0)&CHR$(82)&CHR$(0)&
CHR$(254)
480 B$=B$&CHR$(64)&CHR$(0)&CHR$(0)
490 OUTPUT 10 USING "#,K" ; B$
500 ! ***** ć *****
510 B$=CHR$(139)&CHR$(28)&CHR$(34)&CHR$(0)&CHR$(34)&CHR$(64)&CHR$(162)&CHR$(0
)&CHR$(34)
520 B$=B$&CHR$(0)&CHR$(0)&CHR$(0)
530 OUTPUT 10 USING "#,K" ; B$
540 ! ***** č *****
550 B$=CHR$(139)&CHR$(28)&CHR$(34)&CHR$(0)&CHR$(162)&CHR$(64)&CHR$(162)&CHR$(
0)&CHR$(34)
560 B$=B$&CHR$(0)&CHR$(0)&CHR$(0)
570 OUTPUT 10 USING "#,K" ; B$
580 B$=CHR$(27)&"%"&CHR$(1)&""
590 OUTPUT 10 USING "#,K" ; B$
600 A$=CHR$(13) ! C$=CHR$(10)
610 INPUT B$
620 OUTPUT 10 USING "#,K" ; A$
630 OUTPUT 10 USING "#,K" ; B$
640 OUTPUT 10 USING "#,K" ; C$
650 GOTO 610
660 END
```

Jednostavan program . . .

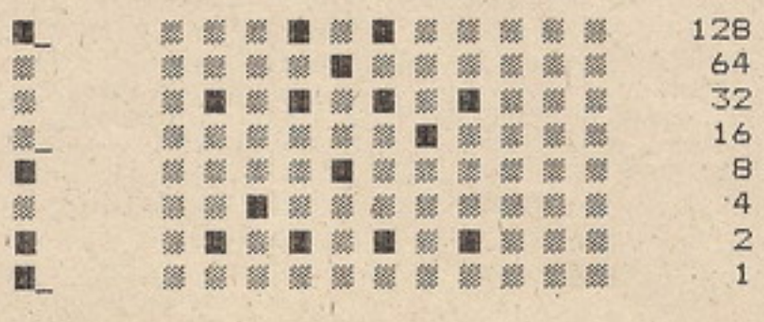
Jednostavnim programom, tj. slanjem odgovarajućih instrukcija štampačima FX (80 ili 100), mogu se izmeniti bilo koji znaci iz njihovog seta znakova. Ovako generisani znaci su istog kvaliteta kao i originalni (normalno, sve to zavisi od ‚dizajnerskih' sposobnosti onoga koji ih kreira). Pošto je u priručniku za štampače serije FX dosta razumljivo objašnjeno kako se mogu promeniti pojedini znaci korišćenjem programskog jezika MBASIC (ovo je provereno na računarima „eple II" i „delta-partner"), dajemo primer programa za računar HP-85 i komunikaciju između njega i štampača FX preko serijskih interfejsa. Ulazni bafer štampača (ne interfejsa) mora biti isključen postavljanjem pina br. 4 na DIP prekidaču SW1 u položaj OFF. Ovim programom se ASCII znači čije su decimalne vrednosti 64, 96, 91-94 i 123-126 zamenjuju našim latiničnim slovima: Ž, ž, Š, Đ, Ć, Č, š, đ, ć, č (respektivno). Parametri, koji jedini određuju izgled slova, ne zavise ni od računara ni od programskog jezika koji ćemo koristiti.

Izgled naših latiničnih slova, dobijenih na ovaj način, vidi se iz listinga programa. S druge strane, primer je koristan i kao ilustracija serijskog komuniciranja računara (u ovom slučaju HP-85) i štampača.

Programske instrukcije na linijama 40 i 50 služe da se serijski interfejs računara HP-85 postavi na format reči od 8 bitova i brzinu prenosa od 1200 (isto kao i u štampaču). Naredbama na linijama 60 i 70 bira se američka azbuka, a linijama 80 i 90 kopira iz ROM-a u RAM izabrana azbuka. Naredbama 100 i 110 se ‚kaže' da će se menjati ASCII znak čija je decimalna vrednost 64. Sada se pristupa izmeni prema objašnjenju na str. 3-79 uputstva za štampače FX. Na primeru našeg malog slova ‚ž' objasnićemo kompletan postupak za kreiranje novog znaka. Donja slika predstavlja crtež za željeno slovo. Leva (odvojena) kolona, čija je decimalna vrednost 139, je tzv. atribut objašnjen na str. 3-80 pomenutog uputstva. Bit najveće težine (vrednost 128) označava, kada je setovan, da se deveti red, red ispod normale (kao kod slova p, q, y), ne koristi, pa zato i nije prikazan na slici. Sledeća tri bita predstavljaju početnu kolonu za štampanje (ovde prva), a preostala četiri krajnju (u našem primeru jedanaesta, a to je decimalni zbir tamnijih tačaka). Kolone iz crteža za samo slovo koriste se za izračunavanje parametara za naredbe na linijama 190 i 200 (to je decimalni zbir vrednosti desnog niza za svaku tamniju tačku). Na primer, četvrta kolona je zbir:

0+2+0+0+0+32+0+128=162

Na sličan način su generisana i ostala naša latinična slova. Ovako se mogu kreirati bilo koji karakteri, uključujući, naravno i ćirilična slova, kao i grafiku. Poslednje dve kolone su rezervisane za razmak, ali se i predposlednja može iskoristiti, na primer, za naša ćirilična slova „lj" i „nj".



Tasteri koje moramo pritisnuti da bi printer odštampao naše slovo zavise od konkretnog računara. Dajemo uporedni prikaz znakova iz američke azbuke, koje smo zamenili našim, i novodobijenih slova:

```
@@@@@@   [[[[[   \\\\\   ]]]]]   ^^^^^^

ŽŽŽŽŽ   $$$$$$   ĐĐĐĐĐ   ĆĆĆĆĆ   ČČĆĆĆ
`````   {{{{{   !!!!!   }}}}}   ~~~~~~

žžžžžž   šššššš   đđđđđ   ćććććć   ččćććć
```

Sada je jasno zašto se u listingu programa na liniji 20 umesto zagrada pojavljuju naša slova "Š" i "Ć".

. . . i „hirurški" rez

Iako se ovakvom metodu malo šta može zameniti, prepravka generatora znakova u štampaču predstavlja mnogo elegantnije i komfornije rešenje. Ovaj način ćemo objasniti na štampaču RX-80, s napomenom da isto važi i za štampače FX-80 i FX--100; razlika je samo u memorijskim lokacijama na kojima su smešteni pojedini znaci. Pretraživanje „epsonovog" ROM-a je predstavljalo dosta mukotrpan posao, ali su rezultati opravdali trud. Prvo smo iz štampača izvadili originalni ROM od 8 K (kompatibilan sa epromom 2764) i izlistali njegov sadržaj na printeru. Sada je trebalo pristupiti najtežem delu posla, pronalaženju memorijskih lokacija na kojima se nalazi sadržaj potreban za generisanje konkretnog znaka.

Trebalo bi, na početku, naglasiti da je ideja bila da se prepravi švedska azbuka (RX-80 ima čak 10 međunarodnih azbuka) tako da američka ili engleska ostanu kom-
```

Tekstovi pisani našim pismima (kako latinicom tako i ćirilicom) mogu se štampati na printeru na dva načina. Prvi se sastoji u realizaciji određenog programa iz računara nakon uključenja štampača, a drugi, znatno elegantniji, u promeni generatora znakova smeštenog u ROM-u samog printera. Ovaj drugi način je posebno interesantan za tip RX-80, kod koga, pošto je znatno jeftiniji, ne postoji mogućnost promene pojedinih znakova programskim putem. Tako štampač RX-80, s obzirom na cenu može da predstavlja privlačniju investiciju, jer se i druga dva nedostatka (niža brzina prenosa i odsustvo valjka za običan tabak) i ne moraju uzeti tako ozbiljno. Za one koji se pitaju zašto uopšte razmatrati prvi način odmah i odgovor: Promena ROM-a zahteva novi eprom tipa 2764 i odgovarajući programator, što je za mnoge, na žalost, ipak nepremostiva prepreka. Zbog toga ćemo se prvo zadržati na programskoj promeni pojedinih znakova.

pletne, pošto je većina računara predviđena za njihov set znakova. Kasnije bi se, prema potrebi, inicijalizovala „švedska" tj. naša azbuka ili programskim putem ili pomoću DIP prekidača SW1 (pinove 6, 7 i 8 postaviti u položaje OFF, ON, OFF, respektivno).

Vratimo se glavnom problemu. Originalni ROM smo prepisivali deo po deo u naš ROM sve dok SELF TEST štampača nije izbacio prve znake. Srećom, to su baš bila specifična slova švedske azbuke, pošto smo nju hardverski (prekidačima) i odredili. Sada je trebalo lokalizovati memorijske adrese. U neke lokacije smo upisali sadržaj 00 da bismo videli kakav će efekat imati na štampanje ili na izgled znaka. SELF TEST je pokazao da su neke kolone ispunjene svim tačkicama, što je dalo zaključak da bitovi (u heksadecimalnom paru za tu lokaciju) čija je vrednost logička 0 daju jednu tačku, dok bitovi logičke vrednosti 1 ne daju ništa. Znači, logika je negativna.

Dužim eksperimentisanjem (uvek je ponovo trebalo prepravljati novi ROM) našli smo da ROM printera RX-80 ima 9 lokacija potrebnih za generisanje nekog znaka. Takođe, potpuno smo identifikovali od koje do koje lokacije se nalazi svaki znak. Na memorijskim adresama

1700-181F

nalaze se specijalna slova raznih azbuka (videti u uputstvu za RX-80 dodatak „G" — fond znakova). Italik ili alternativna forma istih znakova ("kosa slova") nalazi se na lokacijama

1880-1C9F.

Ovakvih specifičnih znakova ima ukupno 32+32. Obična alfanumerika (prve dve stranice pomenutog dodatka) nalazi se na lokacijama:

1820-1B7F

i to poređana istim redom kao u dodatku. Italik forma ovih znakova je na mestima:

1CAO-1FFF.

Njih ima 96+96, što ukupno iznosi 288, uključujući i 32 grafička simbola. Ovo je zaista prebogat fond (čak više nego u štampačima tipa FX).

Prema slikama u dodatku G počeli smo da oblikujemo slova koja su nas interesovala. Posle prvog uspešnog, ostala su išla po uspostavljenoj analogiji:

— za svaki znak je potrebno 8 memorijskih lokacija,

• tačka sa slike konkretnog slova se kodira sa "O", a prazno mesto sa "1",

— tzv. međutačke nalaze se u parnim kolonama slike za to slovo.

Pošto, kao što smo videli, RX-80 ima solidnu grafiku, biće najbolje da se postupak za stvaranje novog znaka tako i prikaže.

Na primer, na lokacijama 07FC-1804 (heksadecimalno) nalaze se sledeće vrednosti

FF C3 7F FD FF FD 7F C3 FD.

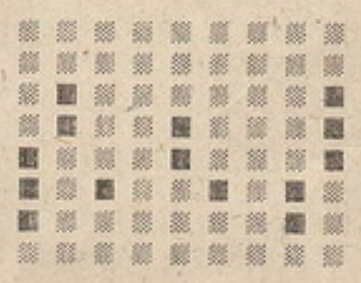
Ove heksadecimalne vrednosti predstavljaju 9 kolona kojima se generiše slovo švedske azbuke "ü". Ako se uzme negativna logika (0 predstavlja tačku u koloni, a 1 prazno mesto), onda će navedeni niz dati sliku tog karaktera u matričnom obliku 9×8. Međutim, ukoliko se u date lokacije upišu vrednosti:

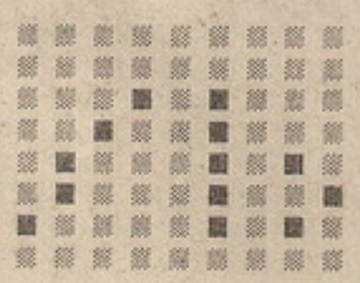
E3 DD FF 5D BF 5D FF DD FF,

onda se umesto tog znaka dobija naše latinično slovo č, a vrednostima

FF CF F7 FF F7 FF F7 FF C1

isto, ali ćirilično slovo. Bitovi za jednu kolonu po težini opadaju odozgo nadole (bit najveće težine je na vrhu). Tako se dobijaju sledeće slike pomenutih znakova:





Kao što se vidi, na ovaj način se može generisati proizvoljan znak koji će biti odštampan na printeru ukoliko mu se pošalje odgovarajući ABCII kod (kod za slovo koje je zamenjeno u ROM-u).

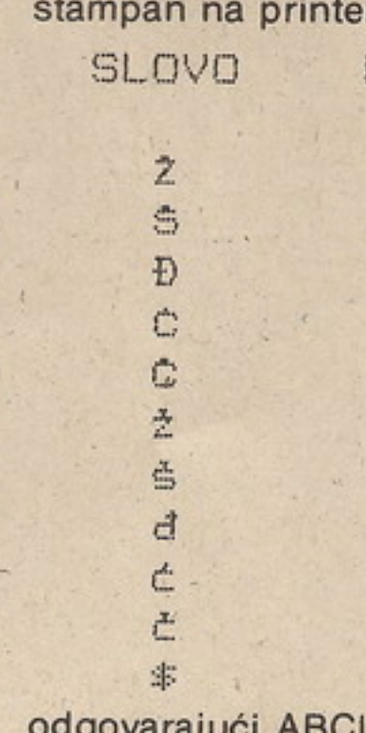
Drugi primer ilustruje šta se dobija kada se u ROM-u štampača, na lokacijama 1FAF-1FB7, vrednosti

F1 CF FB FF E7 FB FF F9 C7

koje predstavljaju slovo 'w' u tzv. italik formi, zamene vrednostima

FD F3 EF DF FF C1 FF F5 FB

čime se dobija naše ćirilično slovo 'lj' prikazano na sledećoj slici:



Slova u ovoj formi ne izgledaju baš najlepše, ali je njihov izgled, kada se odštampaju, sasvim zadovoljavajući.

Na kraju, evo i početnih lokacija, kao i sadržaja koji bi, počev od tih lokacija, trebalo upisati u ROM za sva naša specifična slova:

SLOVO	MEM.LOK.	SADRŽAJ
ž	1805	FF BD FB 35 67 2D DF BD FF
š	17CF	DB AD FF 2D 7F 2D FF AD F3
Đ	17D8	EF 7D 01 FF 6D FF 7D BB C7
Č	1775	C3 BD FF BD 7F BD FF BD DB
Ć	17E1	C3 BD FF 3D 7F 3D FF BD DB
ž	180E	DD FB 5D B7 5D EF DD FF FF
š	17EA	EF D5 FF 55 BF 55 FF D5 FB
đ	17F3	F3 ED FF ED FF AD FF 01 BF
ć	177E	E3 DD FF DD BF 5D FF DD FF
č	17FC	E3 DD FF 5D BF 5D FF DD FF
$	1763	DB FF AB FF 01 FF AB FF B7

Date su vrednosti i znak "$", pošto njega nema u švedskoj azbuci. Što se tiče printera tipa FX, kod njih je za svaki znak u ROM-u predviđeno 12 mesta: 1 atribut, 9 za znak i 2 za razmak. Ukoliko bismo želeli da ROM promenimo i kod ovih štampača, evo i početnih lokacija u koje bi trebalo upisati sadržaje koji su već navedeni:

SLOVO :	ž	š	Đ	Č	Ć	ž	š	đ	ć	č	$
FX-80 :	132D	12E5	12F1	126D	12FD	1339	1309	1315	1279	1321	1255
FX-100:	132A	12E2	12EE	126A	12FA	1336	1306	1312	1276	131E	1252

Ćirilična slova sa printera RX-80 (upisana u ROM-u umesto karaktera u italik formi) imaju sledeći oblik:

ЖАБЦДЕФГХИЈКЛМНОП РСТУВЉЊЏЗШЂЋЧ_

жабцдефгхијклмноп рстувљњџзшђћч

Nadamo se, a to nam je i bila želja, da će ovaj članak zadovoljiti interesovanje onih čitalaca koji već imaju neki od štampača „epson" i, s druge strane, pomoći onima koji tek planiraju da svom računaru dodaju, ipak, neophodan štampač.

Srboljub Kuzmanović

ukleti dvorac i druge bajke

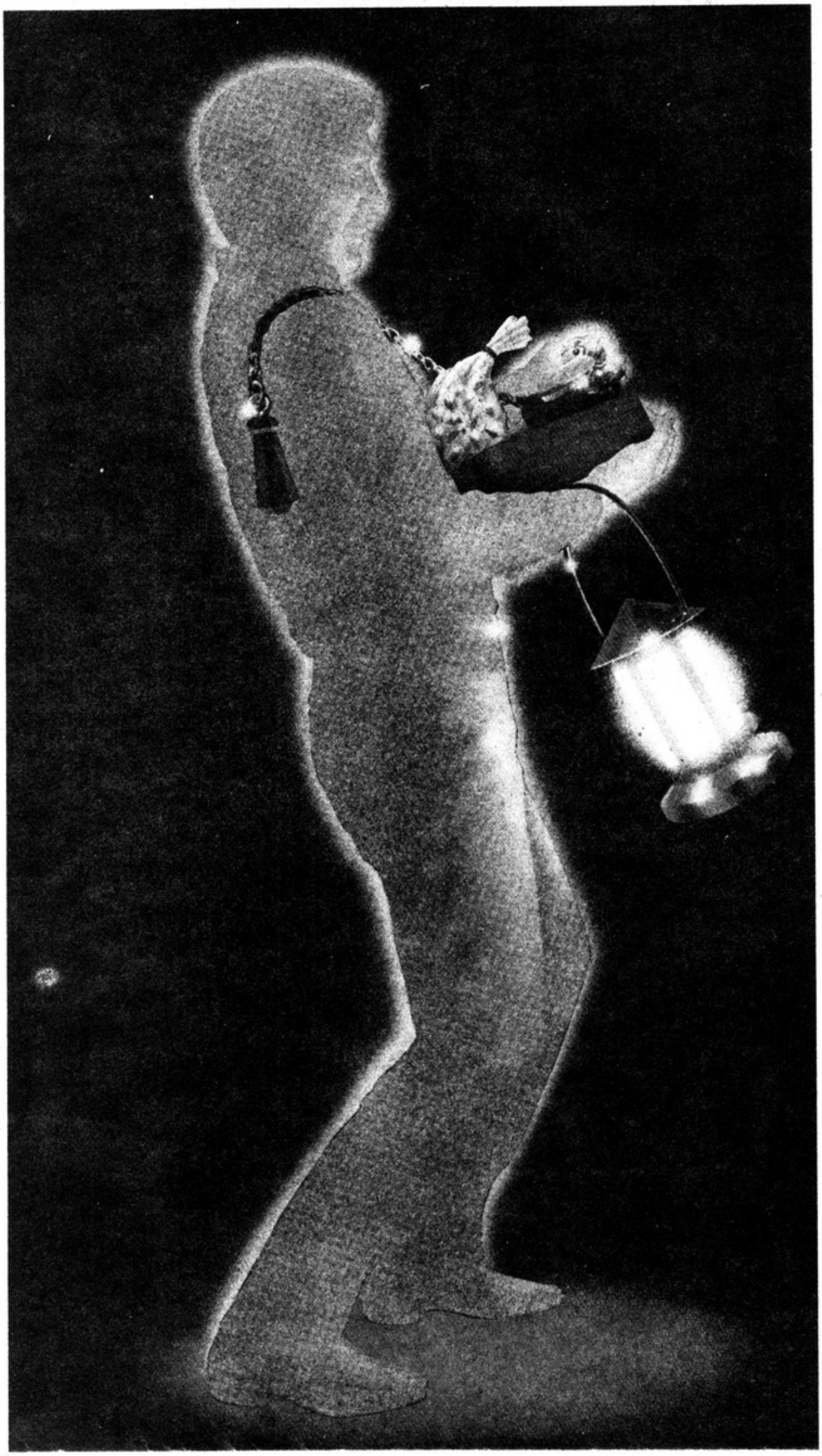
Škola avanturističkih igara

Ako kupite neku avanturu koja košta desetak funti i pokušate da je izlistate, čekaće vas ne malo iznenađenje: program je u celini pisan na bejziku! Zar strane softverske firme nemaju dovoljno poznavalaca mašinskog jezika? Naravno da imaju, ali se pokazalo da je pisanje igre avanture na mašincu, nepotrebno razmetanje u ovim igrama ne dominira brzina već domišljatost i logika — računar razmatra i analizira mnogo različitih situacija i reakcija igrača, što se na nekom višem programskom jeziku (u našem slučaju bejziku) obavlja daleko lakše nego na jezicima nižeg nivoa. Avanture, dakle, mogu da pišu i oni koji se užasavaju „pikovanja" i „poukovanja" po memoriji svog kompjutera.

Pisanje igara avantura je daleko bolja škola za profesionalnog programera nego što bi se reklo na prvi pogled. Prilikom pisanja nekog poslovnog paketa za obradu podataka ili pogotovu, kompajlera nekog programskog jezika, programer treba da predvidi sve moguće ulaze (kako tačne tako i pogrešne) i pobrine se da računar uvek reaguje na pravi način. Ovakav zanat može dobro da se „ispeče" pisanjem avantura u kojima se neće javljati 'Type mismatch', 'Nonsence in BASIC' ili neka slična poruka svaki put kada igrač otkuca POJEDI HELIKOPTER.

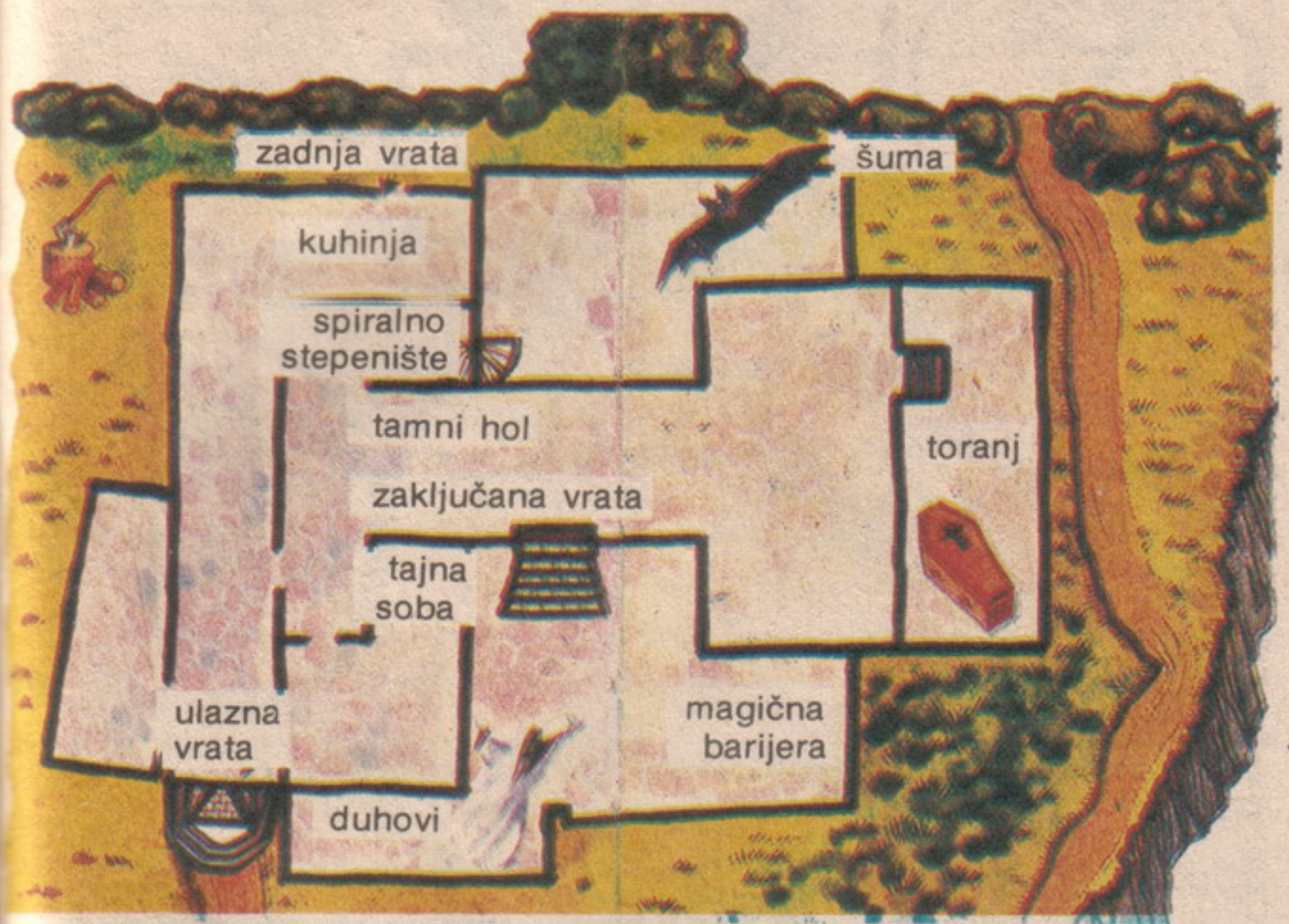
U toku ove škole zajednički ćemo napisati jednu umereno jednostavnu igru — avanturu koja se zove „ukleti dvorac". Ukoliko do sada niste imali kontakt sa igrama ovoga tipa (što je krajnje neverovatno) ili niste uspeli da nađete zajednički jezik sa kompjuterom koji govori jedino engleski, otkucajte najpre program i provedete izvesno vreme otkrivajući tajne ove igre. Dobrim poznavaocima bejzika bismo predložili da ubede (kako li se to radi?) nekoga drugoga da im otkuca ovaj program, jer će, u protivnom, i nehotice otkriti neke od tajni i igra će izgubiti na draži.

„Ukleti dvorac", pisan na standardnom majkrosoftovom bejziku, zanima oko 8 kilobajta korisničke memorije i zahteva još otprilike toliko za nizove i promenljive (vlasnici „spektruma 48 K" će morati da modifikuju program prema sugestijama koje dajemo na kraju teksta) dok druge komercijale avanture umeju da zauzmu i po pedesetak kilobajta. Ipak, najgori način da pišete program ovoga tipa je da sednete, uključite kompjuter i pomislite: „Teško meni, treba da napišem šesnaest kilobajta programa". Savetujemo vam da se za početak ne približavate računaru — uzmite olovku i papir i pokušajte da zamislite osnovni scenario vaše igre.



U potrazi za tajanstvenim predmetima: Nijedna avanturistička igra ne može se okončati bez punog naramka predmeta koji igraču daju naročitu moć

U „Računarima 2" smo posvetili dosta prostora onima koji žele da pišu akcione igre. Ovoga puta se, prirodno, obraćamo onima koji su, najpre, utrošili silne dane da provedu Binga kroz Wilderland, a zatim i nekoliko časova lupajući glavu „kako li je ovaj program napisan;". Pisanje igara avantura je daleko jednostavnije od pisanja akcionih igara — potrebno je samo solidno poznavanje bejzika, posebno rada sa alfanumericima. Kod avantura je, međutim, obilje dobrih i originalnih ideja daleko potrebnije nego za pisanje akcionih igara. Avanture su, dakle, prava stvar za maštovite početnike!



Okvirni plan „Ukletog dvorca"

Pravila igre

Vreme u koje se smeštaju igre avanture je najčešće daleka prošlosti ili daleka budućnost. U poslednje vreme se javljaju i avanture iz svakodnevnog života, kao što je „Kontrabant". Kontrabant je prva ozbiljna avantura pripremljena kod nas i deo je kasete sa programima za računar „spektrum" koju izdaje Radio Student. Tema je u, pravom smislu, savremena: kako uvesti računar u Jugoslaviju. Osnovni razlog za zadržavanje u dalekoj prošlosti su nezaobilazni magični rekviziti koji se vrlo dobro slažu sa igrama ovoga tipa. Priznaćete da je prilično teško tražiti od nekoga da ozbiljno shvati igru u kojoj se ispred Beograđanke pronalazi čarobni štap ili zarobljena princeza koju čuva vukodlak!

Pošto smo odredili istorijski trenutak u koji stavljamo našu avanturu, treba da odlučimo o ciljevima koje igrač treba da postigne. Obično se od njega traži da pobegne sa nekoga mesta, ili da se vrati negde, ili da sakupi neke vredne predmete, ili da oslobodi princezu.

Prema zadacima koje igrač treba da obavi, pripremamo okvirnu mapu mesta događaja, u našem slučaju začaranog dvorca. U ovom trenutku mapa je daleko od detaljne; potrebno nam je samo da uočimo osnovni raspored prostorija koje su bitne za igru, a docnije ćemo dodati i mnogo „slepih hodnika". Okvirna mapa začaranog dvorca i okoline data je na prvoj slici.

Pošto ste nacrtali okvirnu mapu, treba da izaberete podatke koje ćete saopštiti igraču na početku igre. Tih nekoliko rečenica treba da budu takve da probude u njemu interes za avanturu, i da mu, eventualno, na skriven način, kažu nešto o zadacima koje treba da obavi. Evo onoga što treba da zna čovek koji se usudi da priđe začaranom dvorcu.

„Ukleti dvorac se nalazi blizu samoga kraja sveta, okružen močvarom koja je na ivici ogromne i strašne provalije. Njegovi krivi i padu skloni tornjevi se jezivo pružaju ka uvek tamnom i oblačnom nebeskom svodu. Ni jedan zid, ni jedan ugao nije prav, ni jedan prozor nije čitav, ali ni kroz jedan ne dopire nikakva svetlost. Ni malo čudno, niko ne živi u dvorcu ili, da budemo precizniji, ni jedan čovek ne živi u njemu. . .

Priča se da je, u nekoj dalekoj prošlosti, najbogatiji čovek na svetu, bežeći od svojih naprijatelja, proveo poslednje dane u dvorcu, plaćajući zlatom gostoprimstvo zlih sila koje su mu prodale predmete neophodne za opstanak. Začudo, niko nikada nije video njegovo telo. . .

Crtanje mape

Sledeći korak je crtanje precizne mape dvorca. Najlakše ćemo je dobiti ako najpre nacrtamo matricu sa onoliko polja koliko želimo da imamo lokacija (u našem slučaju se neke lokacije odnose i na okolinu dvorca), a zatim svako polje označimo brojem počevši od nule, baš kao na slici 2.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Matrica za ucrtavanje prolaza

Na matrici, koristeći običnu olovku, ucrtavamo zidove i vrata. Zatim, u skladu sa njima, obeležavamo svaku prostoriju imenom koje odgovara našem scenariju. Mi smo izabrali imena kao što je „ugao sveta", „soba puna paučine", „dubok podrum. . .", ali i sasvim ovozemaljske nazive kao što su „kuhinja", predsoblje", „biblioteka". . . Osim imena prostorija, treba da zamislite načina na koje se iz jedne sobe stiže u drugu, tj. da rasporedite vrata. Neka vrata mogud a budu i jednosmerna, ali za to morate da smislite jak razlog. U našoj avanturi se, na primer, ulazna vrata zamka zatvaraju kada prođete kroz njih čime „personal dvorca" sprečava kukavice da pobegnu pre nego što padnu u šake zlim silama. Moguće pravce kretanja je najlakše označavati stranama sveta, pri čemu je sever na gornjem kraju mape. Obično se koriste engleske skraćenice N, S, E, W za sever, jug, istok i zapad respektivno. U ukletom dvorcu je moguće koristiti kako engleske skraćenice, tako i naše rečenice „idi na sever", „idi na zapad" i slično.

Ubacivanjem stepeništa podrumskih vrata i tavanskih prozora kojima se stiže iz sobe u sobu, stvara se utisak trodimenzionalnosti. Kada završite ovu školu možete, naročito ako posedujete računar sa RAM-om solidnih „dimenzija", da pokušate da napišete pravu trodjmenzionalnu avanturu, dodajući, na primer, podrum u pokrovlje našem dvorcu. Na ovim dodatnim nivoima obavezno se nalazi veliki broj soba, približno jednak broju soba u prizemlju. Vredne predmete ćete, naravno, sakriti u najudaljenije budžake poslednjeg sprata.

Specijalne efekte možete da dobijete dajući isto ime različitim lokacijama. U avanturi „Jerusalem" koja je pisana za TRS 80 četiri objekta nose naziv ULICA JERUSALEMA. Da biste se „ispetljali" iz ove ulice morate, najpre, da znate u kome se od

njenih komponenti nalazite, a zatim da primenite neki algoritam kretanja. Pošto smo se trudili da „Ukleti dvorac" ne bude preteška avantura, nismo u nju ubacili zagonetke ovoga tipa.

Sledeća tačka „dnevnog reda" su blaga koja treba smestiti u prostorije. Ova blaga mogu da budu prave dragocenosti poput zlata i dijamanata, ili tajanstveni planovi i dokumenti ili, jednostavno, tragovi koji pomažu da se reši neka zagonetka. Osim objekta koji mogu da se ponesu, treba uključiti i ormare, stolove, vrata i sličnu „nepokretnu imovinu". Neki od ovih predmeta mogu da imaju fioke, pregrade i slične delove, koje igrač najpre treba da ispita da bi ih iskoristio, dok drugi mogu da budu samo lažni tragovi. U našoj avanturi se, na primer, u ormanu nalazi kaput na kome treba pronaći džep u kome je ključ.

Pre nego što konačno rasporedite blaga, treba da zamislite i prepreke koje će ih čuvati: avantura u kojoj bi igrač jednostavno obišao sve prostorije i pokupio, blaga bi bila previše jednostavna. Neka blaga treba da budu čuvana posebno pažljivo i da zahvataju posedovanje određenog broja predmeta da bi se osvojila. U programu „Sphinx adventure", koji je pisan za BBC B, jedno od blaga čuva slon. Da bi se slon oterao, treba dovesti miša, a miša, opet, treba uhvatiti pomoću sira i mišolovke koje treba najpre pronaći.

U gornjem primeru sir i mišolovka nisu nikakvi posebno vredni objekti; njih igrač treba da pronađe da bi rešio neku zagonetku, a zatim može da ih baci. Možete da odlučite da dodate i neke naoko vredne i korisne ali inače sasvim napotrebne predmete, iako ovako nešto nije u skladu sa „fer plejom".

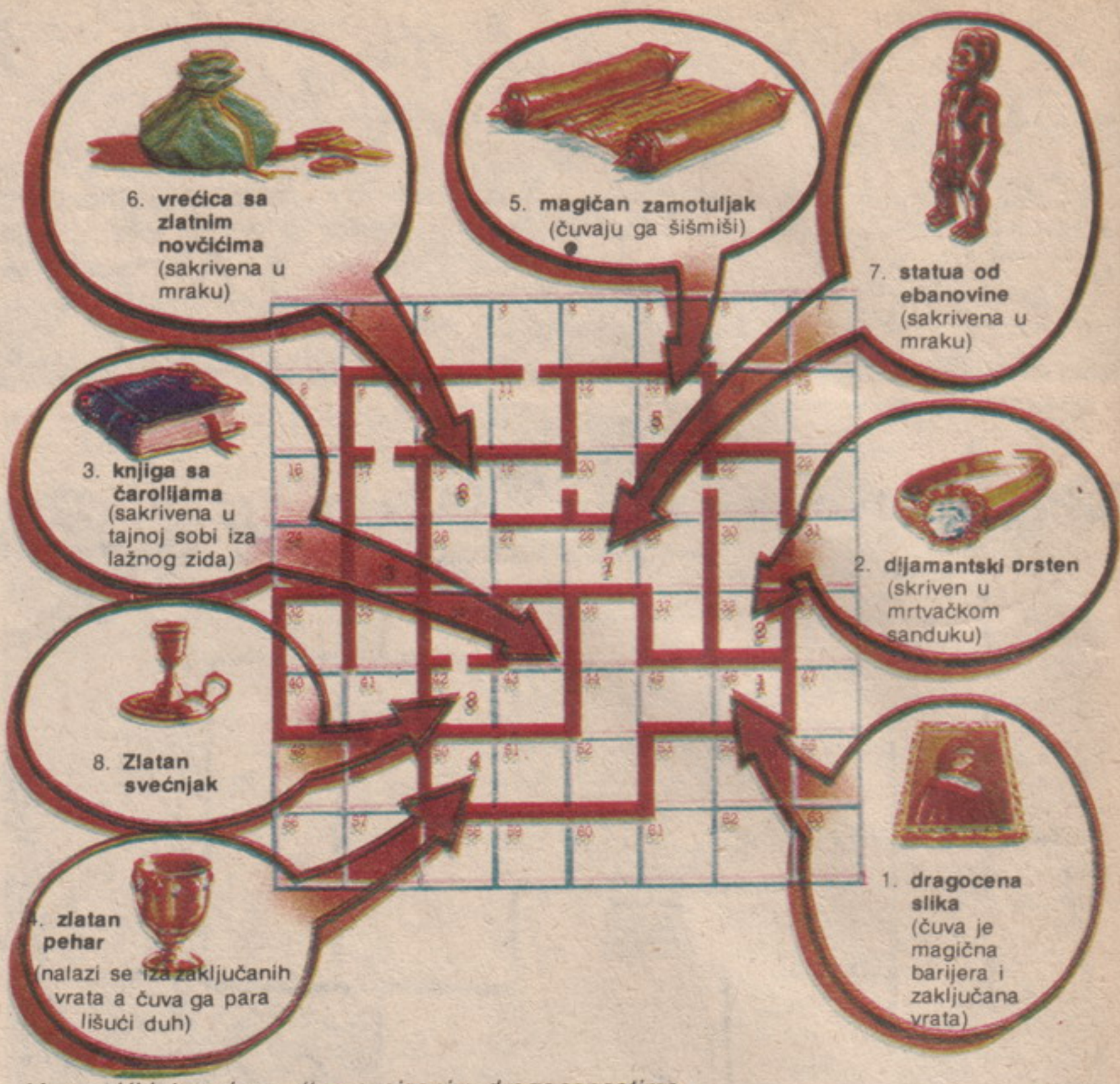
Na slici 3. je prikazana mapa „Ukletog dvorca" i okoline na kojoj su ucrtani zidovi, pravci kretanja i blaga koja igrač treba da pronađe. Blaga su, kao što vidimo, oveležena brojevima 1—8 (nešto docnije ćemo videti zašto ni jedno blago nije označeno brojem 0) i neka od njih su čuvana na odgovarajući način, koji je takođe ubeležen.

Korisni predmeti i zagonetke

Neka od blaga mogu, osim same vrednosti, imati i neku sekundarnu funkciju. U našoj avanturi svećnjak je, najpre, potreban da bi se upalila sveća, ali predstavlja i „blago" bez koga igra ne može da se završi. Ostale korisne objekte treba da upišemo na spisak dragocenosti numerišući ih brojevima 9, 10, 11... Igru ćemo otežati uvođenjem predmeta koji nemaju očiglednu funkciju: umesto ključa, možemo, na primer, negde da sakrijemo zlatnu ukosnicu. Važno je, međutim, da je upotreba predmeta opšte poznata, makar i iz crtanih filmova: u gornjem primeru nije bilo važno da li se slonovi stvarno boje miševa ili ne.

Predmete treba, takođe, smestiti na logična i pristupačna mesta. Ponekad se, u nepažnji, dogodi da smestimo ključ u sobu koju čuva aždaja. Nju treba ubiti mačem koji je sakriven u kovčeg, a kovčeg ne možemo da otvorimo bez ključa. Ovakve greške ćemo, doduše otkriti u fazi testiranja programa, ali su sada ispravke daleko teže.

Sledi izrada spiska zagonetki koje igrač treba da reši. Pošto je „Ukleti dvorac" jednostavna avantura, pripremili smo za nju samo devet zagonetki koje su ispisane na slici 4. Na kraju teksta ćemo napisati nekoliko reči o eventualnim proširenjima programa koja se uglavnom svode na dodavanje novih problema.



Mapa „Ukletog dvorca" sa upisanim dragocenostima

slika 4: Problemi i rešenja

Problem	Rešenje	Potrebni objekti
Previše mračno	Upali sveću	Sveća, svećnjak, šibice
Napadaju šišmiši	Pošpricaj šišmiše superbumom	Superbum sprej
Tajna soba	Probij lažni zid	Sekira
Zaključna vrata	Otključaj ih	Ključ
Parališući duh	Usisaj ga	Usisivač, baterije
Magična barijera	Kaži magičnu reč	Magična cedulja
Močvara	Uzmi čamac (posle jedne upotrebe čamac se zaglibi)	Čamac
Zatrpani prozor	Iskopaj rupu	Lopata
Mrtvački sanduk	Otvori ga	ništa

Spisak imenica i glagola

Pošto pripremite kompletan scenario, treba pristupiti formiranju rečnika. Igrač komunicira sa računarom obično pomoću kratkih rečenica, koje se sastoje od glagola i imenice. Ukoliko pišete avanturu kod koje će se komunikacija odvijati na srpskohrvatskom jeziku, u nekim slučajevima je nužno koristiti povratnu zamenicu 'se', a neophodan je poneki prilog. Očekujte, osim toga, i ogromne probleme sa padežima. Zato je zgodno primeniti mali trik koji korisnici avantura retko primete: reči mogu da se prepoznaju samo po prvih nekoliko slova, najčešće tri. Obzirom da je „Ukleti dvorac" školski primer avanture, nismo želeli da se služimo trikovima, pa smo se potrudili da

slika 5: Rečnik 'Ukletog dvorca'

Broj glagola	Glagol	Odgovarajuće imenice	Značenje
0	-	-	Glagol nije pronadjen
1	upomoc	-	Prikazuje rečnik
2	inventar	-	Predmeti koje igrač nosi
3	idi	strane sveta	Kretanje
4	n	-	Kretanje na sever
5	s	-	Kretanje na jug
6	w	-	Kretanje na zapad
7	e	-	Kretanje na istok
8	u	-	Kretanje uz stepenice
9	d	-	Kretanje niz stepenice
10	uzmi	predmeti	Uzimanje vidljivog predmeta
11	digni	predmeti	Kao uzmi
12	ispitaj	bilo šta	Ispitivanje skrivenih osobina predmeta
13	otvori	vrata, fioka	Očigledno
14	pročitaj	knjige, cedulja zamotuljak	Prikazivanje nekih tajnih poruka
15	kaži	bilo šta	Očigledno
16	kopaj	rupa	Očigledno
17	upotrebi	lopata, konopac	Probijanje zida ili vešanje
18	popni se	-	Penjanje uz konopac
19	upali	sveća	Očigledno
20	ugasi	sveća	Očigledno
21	pošpricaj	šišmiši	Ubijanje insekata superbumom
22	uključi	usisivač	Usisavanje duhova
23	otključaj	vrata	Očigledno
24	baci	predmeti	Ostavljanje nepotrebnog objekta
25	skor	-	Prikazivanje skora
26	sidji	-	Silaženje niz konopac

izaberemo reči koje će se uklopiti u naše rečenice.

Spisak glagola koje kompjuter poznaje uz imenice na koje se glagoli odnose dat je na slici 5. Glagole smo, kao što se vidi, obeležili brojevima 1, 2,. . . dok je nula rezervisana za situaciju u kojoj traženi glagol nije pronađen na spisku. Uvrstili smo u naredbe UPOMOĆ i INVENTAR, koje, respektivno, daju igraču spisak naredbi koje računar poznaje, kao i spisak predmeta koje trenutno nosi. U složenijim avanturama naredba UPOMOĆ je daleko složenija: u odgovoru na nju računar u nekim slučajevima daje igraču indicije o rešenju nekog problema. Ukoliko ste igrali „Hobbita", svakako znate da je najveća veština u tumačenju odgovora koje računar daje kada otkucate HELP.

Neke reči mogu da imaju isto značenje (npr. uzmi i ponesi), omogućavajući igraču da otkuca rečenicu koja mu prva padne na pamet. Ako te reči tretirate istim potprogramima, možete uštedeti određeni memorijski prostor što znači da ovakve sinonime treba uočiti od samoga početka.

Glavni plan

Na slici 6 je dat konačni plan naše avanture. Vidimo da su sve prostorije označene brojevima, da su označeni svi prolazi i vrata, da je svaki objekat obeležen brojem i smešten na odgovarajuće mesto, te da su upisani razni problemi sa kojima će se igrač suočiti (raster, na primer, označava mračne predele, upisana je magična barijera. . .).

Glavni plan je poslednji trenutak da izmenite ponešto u scenariju avanture; kada ga konačno usvojite, preostaje samo tehnički posao pisanja programa. Na tržištu mogu, čak, da se nađu i programi koji, na osnovu glavnog plana, sami pišu igre avanture! Pomoću jednog takvog programa je, uzgred budi rečeno, pisana čak i igra „Kontrabant". Ovo pominjemo zbog toga što pisanje igre avanture po komplikovanom scenariju početniku može da izgleda nedostižno. Treba, ipak, imati poverenja u sebe: došli smo do mesta od koga bi čak i kompjuter mogao da nastavi dalje!

Smeštanje podataka

Došlo je, najzad, vreme da počenemo da pišemo program. Taj program, najpre, mora da poseduje malu biblioteku podataka u kojoj treba da se nalaze imena svih soba i objekata kao i čitava mapa dvoraca. Za ovakvo smeštanje koristimo nizove i matrice.

Niz podataka možemo da zamislimo kao poštanske fahove. Cedulja smeštena u svakom fahu predstavlja samostalni elemenat koji, ipak, ima logičku vezu sa ceduljama u ostalim pregradama. Čitav niz podataka ima ime (npr. A, B, NIZ i slično), dok se njegovi elementi razlikuju po brojevima (npr. A(0), A(1),. . . , A(N)). Sa N smo obeležili takozvanu dimenziju niza koja označava broj njegovih elemenata. Da bi računar znao koliko memorijskog prostora da rezerviše za neki niz, na samom početku programa koristimo naredbu DIM iza koje pišemo imena svih nizova koje ćemo koristiti i njihove dimenzije.

U „Ukletom" dvorcu se koristi više nizova. Prvi i najlakše shvatljiv je niz D$( ) koji sadrži imena i opise svih prostorija u zamku i okolini. Obzirom da, prema glavnom planu, zamak ima 64 prostorije, koristićemo DIM D$(63) (zašto ne DIM D$(64)?). Imena svih lokacija će biti zapamćena prema redosledu, svojih karakterističnih brojeva.

Potreban nam je, zatim, niz R$( ), čiji će elementi čuvati informacije o mogućim pravcima kretanja. Ukoliko, na primer, R$(0) ima vrednost „SE", iz prostorije broj 0 može da se pođe na jug i na istok. Niz R$( ) ima 64 elementa kao i niz D$( ).

Niz 0$( ) sadrži imenice i ostale reči koje se javljaju iza glagola u naredbama koje kompjuter razume. Da bi olakšali promenu broja elemenata rečnika, uveli smo promenljivu W koja označava broj imenica, a zatim koristili DIM 0$(W), naredbu koju razumeju jedino jezici tipa bejzika. Ukoliko poželite da proširite rečnik računara, jednostavno povećajte W i dodajte par novih DATA naredbi.

Niz V$( ) dopunjava rečnik glagolima. Slično imenicama, uvedena je promenljiva V koja označava broj glagola koje kompjuter razume. Prema našoj ranijoj ideji, V ima vrednost 25, pri čemu se V$(0) koristi za situaciju u kojoj glagol nije prepoznat.

Na prvi pogled bi se reklo da nam nije potrebno više nizova. Podrobnije razmišljanje nas, međutim, upozorava da na neki način treba registrovati lokacije pojedinih predmeta, kao i spisak predmeta koje igrač nosi sa sobom. Jedan od načina da rešimo ovaj problem je uvođenje niza L( ) koji sadrži lokacije pojedinih predmeta i niz C( ) koji sadrži brojeve predmeta koje igrač nosi sa sobom. Ukoliko je broj predmeta koji mogu da se nose G (u našem slučaju G=18), koristićemo DIM L(G), C(G). Naprednije programere upozoravamo na mogućnost da, po cenu izvesnog usporenja programa, uštedimo memorijski prostor za niz C( ) time što bi 0 kao lokacija nekog predmeta označavala da ga igrač poseduje.

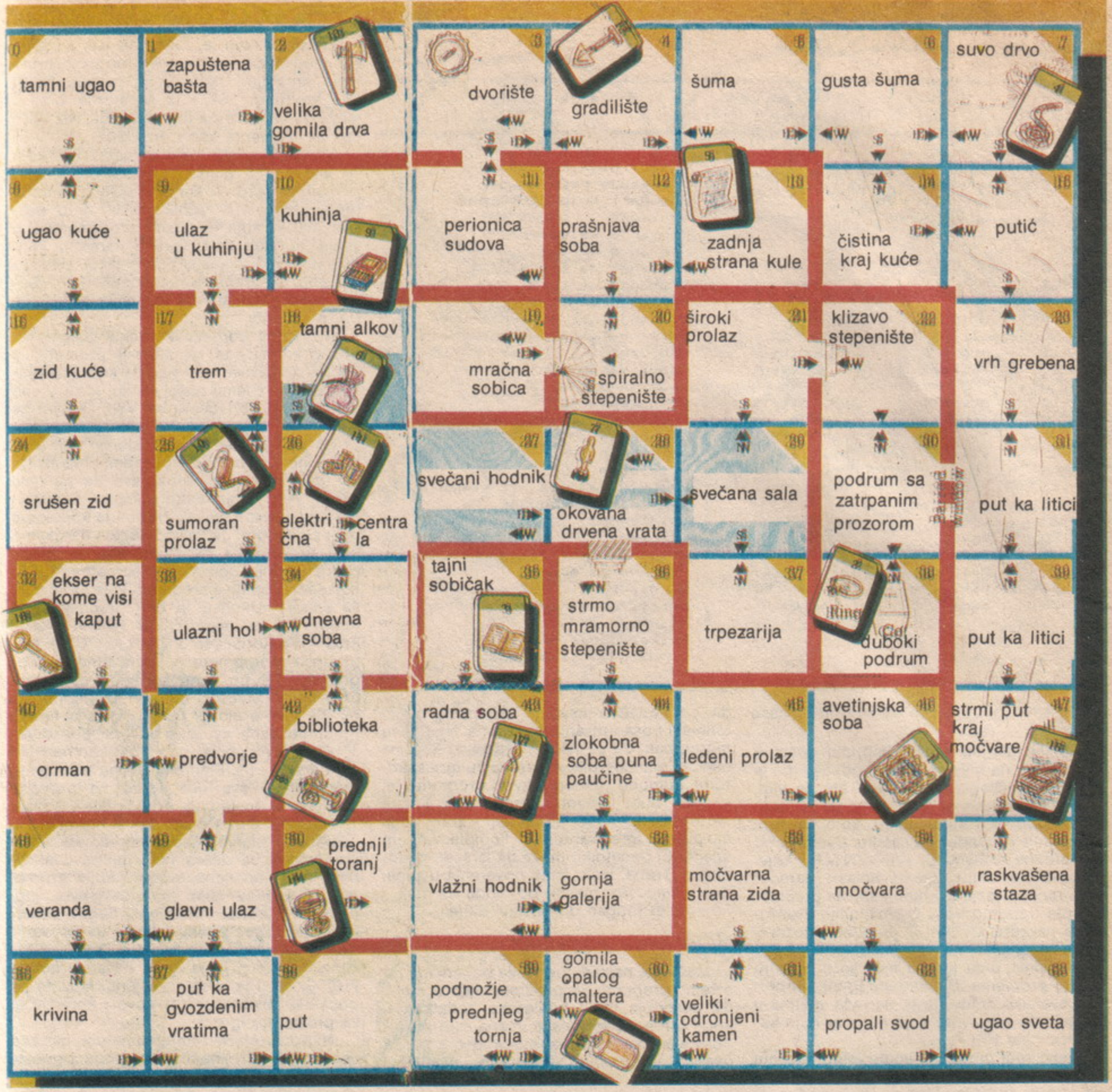
Pored svih pomenutih, potreban nam je i niz flegova F( ). Elementi ovoga niza imaju vrednosti 0 sve dok igrač ne pronađe predmet N, posle čega F(N) dobija vrednost 1. Vrednost 0 označava, dakle, normalno ili neaktivno stanje nekog predmeta, dok vrednost 1 označava neku njegovu aktivnost (npr. upaljena sveća, vidljiv prdmet itd.). Pošto su neke imenice (sever, jug, prsten. . .) jednoznačne, neki flegovi se koriste za „specijalne efekte". Sveći su, s druge strane, potrebna dva flega: F(17) označava da li ju je igrač našao u fioci a F(0) — da li je sveća upaljena. Možete da popišete nekorišćene flegove i koristite ih za proširenje igre.

Najlakši (ali ne i najracionalniji, bar kod većine računara) način da se podaci smeste u nizove je korišćenje READ — DATA naredbi. Da bismo, na primer, preneli nazive lokacija u D$, koristimo naredbe:

```
FOR I=0 TO 63
READ D$(I)
NEXT I
DATA. . .
```

Početno stanje flegova je daleko lakše postaviti. Kada nekom bejzik računaru kažemo da dimenzioniše niz, svi njegovi elementi će imati vrednost nula. Nekoliko flegova (u našem slučaju 2, 17, 18, 23, 26 i 28) u početku treba da dobije vrednost 1, što najlakše postižemo jednostavnim LET naredbama.

Ukoliko posedujemo kompjuter sa manje memorije (na primer ZX 81 sa 16 Kb RAM-a), ili računar koji nema READ — DATA naredbe, moraćemo, najpre, da otkucamo program, a zatim da unesemo podatke i snimimo ih na kasetu zajedno sa samim programom. Na taj način smo izbegli da dugi komentari budu jednom smešteni u DATA naredbu a drugi put u zonu (alfa)



Konačni plan ukletog dvorca

numeričkih promenljivih. Postoje i toliko pametni bejzik interpretatori (jedan takav ima TRS®80) da automatski izbegavaju ovo prenošenje sve dok korisnik ne zatraži promenu nekog na ovaj način unetog podatka i tako štede memorijski prostor i programerovo vreme.

Struktura programa

U skladu sa modernim programskim standardima, razložićemo problem na niz operacija koje kompjuter treba da obavi, a zatim realizovati svaku od njih. Prateći dalji tekst biće neophodno da, s vremena na vreme, konsultujete sekcije samoga programa i tako u praksi proverite ono o čemu govorimo. Posle izvesnog vremena primetiće da se sve lakše snalazite u dugom listingu i da svaka naredba postepeno dobija smisao. Ukoliko ne možete da uočite namenu neke naredbe, možete da pokušate da je izmenite ili čak izostavite i posmatrate rad programa.

Prvi posao koji računar treba da obavi je inicijalizacija. Ona se sastoji od dodeljivanja vrednosti promenljivima i elementima nizova i odvojena je u potprogram koji se nalzi na samom kraju glavnog programa. Zbog čega smo ovo učinili? Pokazuje se da sve GOTO i GOSUB naredbe brže pronalaze linije koje su bliže početku programa. Inicijalizacija se obavlja samo jednom i nema nikakve svrhe da opterećuje početak programa koji će se izvršavati ogroman broj puta u toku igre.

Dolazimo do početka „glavne petlje" u kojoj računar najpre obaveštava igrača o mestu na kome se nalazi. Za to se koristi promenljiva RM u kojoj je stalno smešten broj potrebne prostorije tako da PRINT D$(RM) ispisuje njen opis. Sledi obaveštenje o mogućim pravcima kretanja koje je nešto teže pripremiti. Znamo da su ovi pravci smešteni u promenljivu R$(RM) koju, dakle, treba razdvojiti na slova. To postižemo primenom naredbe MID$: MID$(R$(RM),I,1) izdvaja I-to slovo promenljive R$(RM). Jasno je da I predstavlja indeks petlje koji uzima vrednost 1,2,. . . sve do dužine alfanumerika R$(RM) koju određujemo primenom funkcije LEN.

Treba, posle ovoga, da štampamo imena predmeta koje igrač vidi. To postižemo naredbama:

```
FOR I=1 TO G
IF L(I)=RM AND F(I)=0 THEN PRINT ,,Vidiš"; 0$(I); ,,."
NEXT I
```

Vidimo da računar proverava da li je lokacija objekta I koju čuva promenljiva L(I) jednaka broju prostorije u kojoj se trenutno nalazi igrač (RM), a zatim i da li je predmet vidljiv (F(I)=0). Ukoliko su oba uslova ispunjena, štampa se 0$(I) to jest ime predmeta. Ova naredba se ponavlja za svaki vidljiv predmet, to jest G puta.

Ostaje još da se štampa eventualna poruka koja iskazuje reakciju računara na prethodni igračev postupak. Moguće je, naime, da računar nije bio u stanju da razume ili, u datim okolnostima, izvrši igračev nalog i da treba pružiti obaveštenje o tome. Moguće je, osim toga, da računar saopšti igraču ono što je pročitao u nekoj knjizi ili na nekoj cedulji. Sva ova komunikacija se obavlja pomoću promenljive M$. Pošto je, sa PRINT M$, štampana njena vrednost, računar izvršava M$=,,Šta?", naredbu koja će vam se u prvi mah učiniti čudnom. Malo dublje razmišljanje nas, međutim, uverava u korisnost ove naredbe: ostatku programa se nudi mogućnost da promeni sadržaj promenljive M$ (ukoliko ste sami kucali program, svakako se sećate toga da se većina naredbi završava sa M$=,,. . ."). Ukoliko ni jedna ,,sekcija" programa nije uspela da razume igračevu naredbu i promeni M$, naredba je bila sasvim nerazumljiva ili pogrešno otkucana čemu komentar ,,Šta" odlično pristaje.

Sledeća sekcija programa je izuzetno jednostavna: računar ispisuje ,,Šta da radim", a zatim igračev odgovor smešta u promenljivu Q$ primenom naredbe INPUT.

Faza do koje dolazimo je, za promenu, veoma složena i predstavlja srce čitavog programa: računar treba da analizira Q$, razdvoji je na imenicu i glagol, a zatim da pronađe te reči u svome rečniku i kodira ih njihovim brojevima. Pošto nam je za sada cilj samo da shvatimo kako program radi u globalu, nećemo se više zadržavati na ovoj fazi; docnije ćemo joj posvetiti čitavo poglavlje.

Ukoliko se tokom analize pokazalo da instrukcija nema smisla, treba pripremiti poruku o tome i vratiti se na početak glavne petlje da bi se očekivala sledeća igračeva naredba. Ukoliko instrukcija izgleda logično (programeri bi rekli da je sintaksno ispravna, ali da je njena sementika (značenje) i dalje pod znakom pitanja), računar izvršava potprogram koji će do kraja analizirati i, eventualno, izvršiti naredbu. Obzirom da svaka rečenica koju igrač otkuca treba da počene glagolom, najjednostavnije je čitavu sintaksnu analizu zasnovati na izdvajanju i numerisanju tog glagola brojem između 1 i 26, koji će čuvati promenljiva VB. Zatim ćemo izvršiti ON VB GOSUB n1, n2,. . ., n26 gde je, na primer, sa n2 označen linijski broj bejzik naredbe koja tretira glagol čiji je broj u jeziku koji kreiramo 2.

Pošto je izvršena akcija koja se sastoji u promeni neke od promenljivih ili elemenata nizova i čiji rezultat sumira M$, kontrola se vraća glavnom programu koji proverava da li postoji neka izuzetno prioritetna poruka koju bi trebalo ispisati. Ukoliko, na primer, upalite sveću i nastavite da se krećete sa njom, promenljiva LL će biti umanjivana za 1 posle svakog vašeg poteza. Ukoliko je sveća blizu dogorevanja, računar će vas upozoriti na to. Ovo upozorenje se ne odnosi na vašu prethodnu akciju — ono je, kako bi rekli programeri, ,,sistemske prirode".

Na kraju preostaje samo da se izvrši jednostavna GOTO naredba koja ponovo započinje glavnu petlju. Glavna petlja je, dakle, neka vrsta procedure iz koje se izlazi jedino kada neki od potprograma (to je« potprogram koji realizuje naredbu SKOR) detektuje kraj igre u kom slučaju biva izdata odgovarajuća poruka i izvršena naredba END. ,,Ukleti dvorac" je avantura koja je prijateljski nastrojena prema igraču, pa ne postoji mogućnost da on usput pogine (premda primetiti da igrač može da zapadne u nevolju iz koje nema izlaza bez posedovanja nekog predmeta, što je, na neki način, ekvivalentno sa smrtnim ishodom), ali i sami znate da u većini komercijalnih avantura postoji jedan pozitivan i mnoštvo ,,smrtnih" izlaza iz glavne petlje.

Analiza naredbe

Vraćamo se, kao što smo obećali, trenutku kada je igrač otkucao naredbu koja se sastoji od dve reči (u principu glagola i imenice) koje je računar zapamtio u promenljivoj Q$. Računar, najpre, treba da razdvoji alfanumerik Q$ na reči koje prepoznaje po praznom prostoru kojim su odvojene. To činimo naredbama:

```
10 V$=,,":W$=,,"
20 FOR I=1 TO LEN(Q$)
30 IF MID$(Q$,I,1)=,, " AND V$=,,"
   THEN V$=LEFT$(Q$,I—1)
40 IF MID$(Q$,I+1)<>" " AND V$<>" "
   THEN W$=MID$(Q$, I+1, LEN (Q$)—1):I
   =LEN(Q$)
50 NEXT I
60 IF W$=$" THEN V$=Q$
```

Naredba 30 pokušava da pronađe blanko simbol u alfanumeriku Q$. Ukoliko joj to uspe, promenljiva V$ dobija vrednost prve reči stringa. Potom računar traži prvi sledeći neprazan simbol (za slučaj da su reči odvojene sa nekoliko praznih mesta) i ostatak rečenice smešta u W$. Naredba 60 ima smisla u slučaju da je igrač otkucao samo jednu reč (npr. SKOR). U tom slučaju nije nađen blanko simbol, pa ni V$ ni W$ nisu dobili nikakve vrednosti. Zahvaljujući, međutim, naredbi 60, čitava reč koju je igrač otkucao biva preneta u V$.

Sledi kodiranje (autori bejzik interpretatora bi rekli tokenizacija) igračeve naredbe. Potrebno nam je da oformimo promenljivu VB koja će sadržati redni broj glagola koji je otkucan i smešten u V$. Ništa lakše:

```
10 VB=0
20 FOR I=1 TO V
30 IF V$=V$(I) THEN VB=I
40 NEXT I
```

Program je toliko jednostavan da ga ne treba ni komentarisati. Pokušajte, ipak, da sebi dokažete da ste ga razumeli tako što ćete ga dopuniti programom koji izračunava broj objekta na listi i smešta ga u promenljivu OB.

Ukoliko u spisku reči koje računar poznaje nije pronađena ona koju je igrač otkucao, VB ili OB imaju vrednost nula. Ovakva situacija je dobra indicija za prijavljivanje greške, ali treba primetiti da uspešno kodiranje još ne označava ispravnu konstrukciju, već smo rekli da ispravna sintaksa nužno ne povlači i ispravnu semantiku. Greške tipa ,,OTVORI SVEĆU" će biti otkrivene i prijavljene tek docnije.

Sa prijavljivanjem grešaka, reklo bi se, nema mnogo problema. Zahvaljujući malom triku koji smo primenili dodeljujući promenljivoj M$ vrednost ,,Šta?" na samom početku glavne petlje, bilo bi sasvim dovoljno da program prepozna sve ispravne situacije dok bi neprepoznavanje značilo da M$ ne treba menjati. Ovakvo razmišljanje, međutim, upravo pravi razliku između inventivnog i neinventivnog programera.

Loš program ima jednu ili samo nekoliko poruka o greškama, dok dobar program prepoznaje svakakve situacije i prevazilazi ih. Svaki igrač će mnogo više voleti da mu računar, kada otkuca ,,Ponesi vrata", odvrati sa ,,Što ih ti ne poneseš?" umesto sa ,,Šta!"!

U ,,Ukletom zamku" se obavljaju razni testovi ulaznih podataka. Najpre računar ispituje da li je W$ veće a OB jednako nuli, što označava da je igrač otkucao glagol bez imenice (ova situacija još ima šansi da se povoljno okonča ukoliko je otkucani ,,glagol" SKOR, UPOMOĆ ili INVENTAR, ali računar za sada evidentira potencijalnu grešku komentarom GLUPO koji ne mora nužno biti prikazan). Testira se, zatim, i da li je nepoznat glagol praćen imenicom sa liste; u tom slučaju biva štampano nešto poput ,,Ne mogu da. . ." praćeno rečenicom smeštenom u Q$. Na kraju sekcije testova računar izvršava IF VB=0 THEN VB=V+1. Ova naredba predstavlja neku vrstu sigurnosnog ventila, ali samo za neke računare. Ukoliko VB ima vrednost nula, naredba ON VB GOSUB će izazvati nastavak rada ili ON RANGE grešku (zavisno od kompjutera). Zato smo predvideli jedan nepostojeći glagol čiji je broj V+1 i koji upućuje program na ,,praznu" naredbu RETURN.

Kada je sve provereno, naredba ON VB GOSUB izvršava jedan od potprograma kojim se realizuju naredbe. U okviru tog potprograma biva kompletirana analiza ulaznih podataka, a zatim se nalog izvršava ili biva izdata poruka o grešci. U okviru ovih potprograma treba primetiti mnogobrojne logičke funkcije AND, OR i NOT koje su kao stvorene za ovakve primene.

Već smo pominjali situacije u kojima je nastupio neki nepredviđeni događaj koji zahteva trenutnu pažnju igrača. Ukoliko se, na primer, nađemo u gornjoj kuli, postoji velika šansa da nas napadnu šišmiši. U tom slučaju čak i valjane naredbe bivaju stalno odbijane sve dok ne uništimo dosadne šišmiše. Možete li da zamislite još neku zagonetku ovoga tipa kojom biste dopunili ,,Ukleti dvorac"?

Kretanje po dvorcu

Ostalo nam je još da objasnimo deo programa koji omogućava kretanje po dvorcu i okolini. Kretanje na sever možemo, kao što znamo, da ostvarimo kako sa IDI NA SEVER tako i kucanjem običnog N. Pokušaćemo da objasnimo kako radi deo programa koji čine linije 750-1080.

Najpre se promenljivoj D dodeljuje vrednost koja odgovara pravcu kretanja koji je igrač izabrao (1 za sever, 2 za jug, 3 za zapad, 4 za istok, 5 za gore i 6 za dole). Linija 760 to radi u slučaju da je igrač otkucao samo jedno slovo (N, S, W, U ili D), dok sledećih 6 linija rešavaju slučaj kada su otkucane tri reči (npr. 'idi na sever'). Linije 830-880 su, na neki način, veštački dodane i tretiraju igračevo kretanje gore, odnosno dole, po paru postojećih stepenica. Ova

kretanja se, kao što vidimo, svode na obično kretanje na neku od strana sveta.

Linije 890-360 tretiraju neke specijalne situacije u kojima računar treba da onemogući igračevo kretanje. To kretanje mogu da spreče duhovi, magična barijera, živi pesak u močvari, mračni predeli i težak čamac koji možemo da nosimo samo kratko vreme.

Ukoliko je sve u redu, računar treba da promeni igračevu lokaciju. Taj posao obavljaju linije 970-1040. Analizira se R$(RM), string koji, kao što znamo, sadrži sve moguće pravce kretanja. Ukoliko je pravac kretanja moguć, promenljiva RM dobija novu vrednost i fleg 35 biva postavljen na vrednost 1. Na samom kraju (linija 1050), računar proverava da li je fleg 35 setovan; ukoliko nije, igrač je tražio kretanje u nemogućem pravcu i treba izdati poruku 'Tuda ne mogu'. Linija 1070, najzad, za sva vremena zatvara ulazna vrata kada igrač jednom prođe kroz njih. Izlaz iz zamka je moguć kroz dvorišna vrata, zatrpani prozor i primenom čarolije.

Kako olakšati. . .

Kada razumete kako „Ukleti dvorac" radi, možete da sa modifikujete i učinite jednostavnijim ili složenijim, pojednostavljenju ćete, jasno, pristupiti ako vaš kompjuter nema dovoljno memorije da primi program u sadašnjem obliku.

Vlasnici „spektruma" moraju da modifikuju program da bi uopšte mogao da funkcioniše. Potrebno je, najpre, dodati LET ispred svake naredbe dodeljivanja, što kod Majkrosoftovog bejzika nije nužno. Zatim treba sve alfanumerike u DATA listama (linije iza 1810) omeđiti navodnicima (npr. 1810 DATA „upomoć", „inventar", idi". . .). Otkucajte, najzad, sve programske linije sa slike 7 kojima zamenjujete neke linije glavnog programa. Treba primetiti da je organizacija nizova unekoliko promenjena, pošto kod „spektruma" indeksi nizova počinju od 1 a ne od nule.

I kako otežati igru. . .

Klasično otežavanje igre je uvođenje vremenskog limita. Možete, na primer, da zahtevate da igrač završi posao do ponoći, jer se tada dešavaju grozne stvari. Posle svakoga poteza treba, jednostavno, smanjiti vrednost nekog brojača i ispitati da li je on došao do nule; sličan princip je primenjen i kod dogorevanja sveće.

Možete, zatim, da ograničite broj predmeta koje igrač sme da nosi. Pokušajte da isplanirate ovo ograničenje tako da zadatak može da se obavi samo na jedan način, ali se potrudite da rešenje bude moguće!

Uvedite, zatim, veoma redak događaj (primenom naredbe RND) koji „baca" igrača na neko zabačeno mesto u zamku i lišava ga nekih predmeta koje skriva na stara ili neka nova mesta. Možete da dodate i zatvorske ćelije koje zadržavaju igrača izvesno vreme pod pretnjom oduzimanja nekih dragocenih predmeta. Ima, naravno, mesta i za dodavanje drugih stanovnika zamka i problema sa njima.

Unapredite sistem bodovanja tako da dragoceniji predmeti donose više poena, kao i kazne koje se sastoje od oduzimanja poena u nekim situacijama. U poslednjem slučaju potrebno je obezbediti i način na koji igrač može da „zaradi" oduzete poene i završi igru.

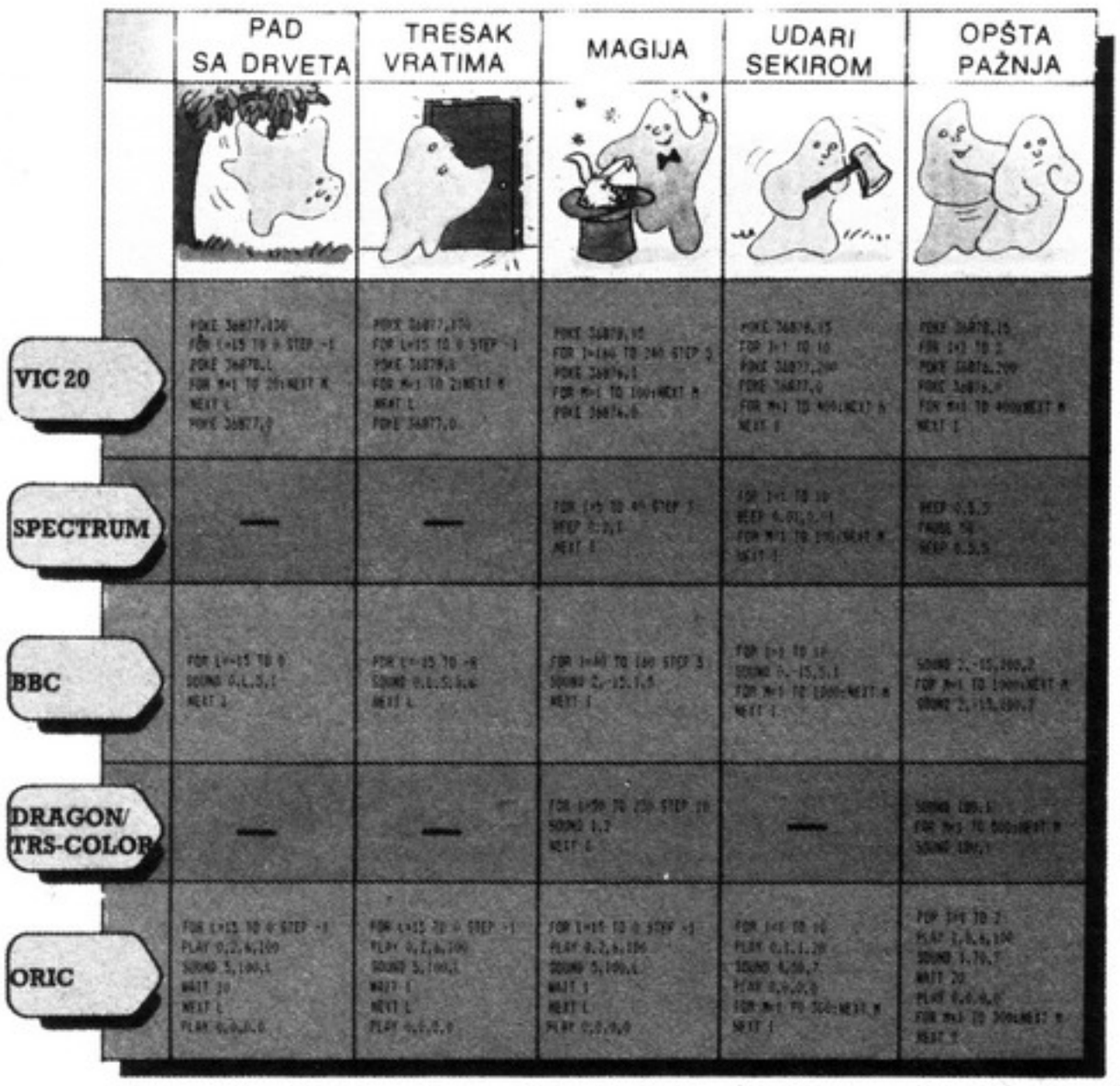
Dodajte, najzad, mogućnost da se u nekom trenutku situacija (sadržaji nizova i promenljivih) snimi na traku kako bi se igra sutrađan nastavila. Rešavanje ovoga problema može korisno da posluži kao motivacija za upoznavanje načina na koji vaš kompjuter opšti sa kasetofonom!

Na slici 8 je, najzad, dato nekoliko naredbi koje dodaju ton u raznim situacijama. Avanturu ćete, takođe, učiniti znatno interesantnijom ako dodate neke grafičke efekte.

Dejan Ristanović

```
 10 REM
 20 REM
 30 REM
 40 REM  ***************************
 50 REM  *       Zli dvorac        *
 60 REM  ***************************
 70 REM
 80 REM   Verzija za  ZX Spectrum
 85 REM
 90 REM  "Racunari u vasoj kuci" 3
100 REM
110 REM
160 LET F$="                    "
220 PRINT D$(RM+1)
240 FOR I=1 TO LEN(R$(RM+1))
250 PRINT R$(RM+1)(I TO I);",";
340 LET X$="":LET W$="":LET VB=0:LET OB=0
350 FOR I=1 TO LEN(Q$)-1
360 IF Q$(I TO I)=" " AND X$="" THEN LET X$=Q$( TO I-1)
370 IF Q$(I+1 TO I+1)<>" " AND X$="" THEN LET W$=Q$(I+1 TO)
    :LET I=LEN(Q$)-1
390 IF W$="" THEN LET X$=Q$
395 IF LEN(X$)>LEN(V$(1)) OR X$="" THEN GOTO 425
396 LET X$=X$+F$( TO (LEN(V$(1))-LEN(X$)))
410 IF X$=V$(I) THEN LET VB=I
425 IF W$="" OR LEN(W$)>LEN(O$(1)) THEN GOTO 470
426 LET W$=W$+F$( TO (LEN(O$(1))-LEN(W$)))
505 IF OB=0 THEN GOTO 530
540 IF RM=44 AND INT(RND*2)=1 AND F(24)<>1 THEN LET F(27)=1
550 IF F(20)=1 THEN LET LL=LL-1
560 IF LL<1 THEN LET F(20)=0
570 GOSUB 610*(VB=1)+680*(VB>2 AND VB<10)+1090*(VB=10 OR VB=11)
    +1150*(VB=12)+1190*(VB=13)+1260*(VB=14)+1300*(VB=15)
    +1340*(VB=16)+1370*(VB=17)+1420*(VB=18)+1470*(VB=19)
    +1520*(VB=20)+1540*(VB=21)+1570*(VB=22)+1600*(VB=23)
    +1630*(VB=24)+1650*(VB=25)+1450*(VB=26)+1740*(VB=27)
920 IF (RM=26 AND F(20)=0) AND (D=1 OR D=4) THEN
    LET M$="Sasvim je mracno":RETURN
950 IF (RM>26 AND RM<30) AND F(20)=0 THEN
    LET M$="Ja u mrak ne idem!":RETURN
960 LET F(35)=0:LET RL=LEN(R$(RM+1))
980 LET U$=R$(RM+1)(I TO I)
1070 IF RM=41 AND F(23)=1 THEN LET R$(50)="SW"
    :LET M$="Vrata su se zalupila":LET F(23)=0
1090 IF OB>6 OR OB=0 THEN LET M$="Ne mogu da uzmem "+W$:RETURN
1310 IF C(3)=1 AND OB=34 THEN LET M$="** CAROLIJA **"
    :IF RM<>45 THEN LET RM=INT(RND*64)
1350 IF C(12)=1 AND RM=30 THEN LET M$="Iscupao sam gomilu greda"
    :LET D$(RM+1)="Kraj otvora u zidu":LET R$(RM+1)="NSE"
1400 IF OB=13 AND C(13)=1 AND RM=43 THEN LET R$(RM+1)="WN"
    :LET D$(RM+1)="U radnoj sobi sa skrivenim separeom"
    :LET M$="Probio sam tanki zid"
1500 IF OB=17 AND C(17)=1 AND C(9)=1 AND C(8)=1 AND LL>1 THEN
    LET M$="Jeste da zmirka ali bar svetli!":LET F(20)=1
1520 IF F(20)=1 THEN LET F(20)=0:LET M$="Ugasena je."
1610 IF RM=28 AND OB=25 AND F(25)=0 AND C(18)=1 THEN
    LET F(25)=1:LET R$(RM+1)="SEW"
    :LET D$(RM+1)="Kraj velikih otvorenih vrata"
    :LET M$="Kljuc se okrece..."
1750 DIM R$(64,4):DIM D$(64,4):DIM O$(W,13):DIM V$(V,9)
1930 FOR I=1 TO 64
2120 FOR I=1 TO 64
2250 REM
2260 REM                                  Dejan Ristanovic 1984.
```

Prilagođenje programa za računar „spektrum"



Zvučni efekti

```
 10 REM
 20 REM
 30 REM
 40 REM  ************************
 50 REM  *     Ukleti dvorac     *
 60 REM  ************************
 70 REM
 80 REM
 90 REM  "Racunari u vasoj kuci" 3
100 REM
110 REM
120 REM prema programu 'Haunted House'
130 REM  Jenny Tyler i Lesa Howartha
140 REM
150 REM
160 REM
170 V=26:W=36:G=18
180 GOSUB 1750
190 CLS:PRINT "Ukleti dvorac"
200 PRINT "--------------"
210 PRINT "Nalazim se"
220 PRINT D$(RM)
230 PRINT "Pravci kretanja:";
240 FOR I=1 TO LEN(R$(RM))
250 PRINT MID$(R$(RM),I,1);",";
260 NEXT I
270 PRINT
280 FOR I=1 TO G
290 IF L(I)=RM AND F(I)=0 THEN PRINT "Vidim ";O$(I);"."
300 NEXT I
310 PRINT "========================="
320 PRINT M$:M$="Sta?"
330 INPUT "Sta da radim";O$
340 V$="":W$="":VB=0:OB=0
350 FOR I=1 TO LEN(O$)
360 IF MID$(O$,I,1)=" " AND V$="" THEN V$=LEFT$(O$,I-1)
370 IF MID$(O$,I+1,I)<>" " AND V$<>"" THEN W$=MID$(O$,I+1,LEN(O$)-1):I=LEN(O$)
380 NEXT I
390 IF W$="" THEN V$=O$
400 FOR I=1 TO V
410 IF V$=V$(I) THEN VB=I
420 NEXT I
430 IF W$="se" OR VB=26 THEN W$="konopac":OB=14:GOTO 470
440 FOR I=1 TO W
450 IF W$=O$(I) THEN LET OB=I
460 NEXT I
470 IF W$>"" AND OB=0 THEN M$="Glupo!"
480 IF VB=0 THEN VB=V+1
490 IF W$="" THEN M$="Recenica obicno ima subjekat i predikat"
500 IF VB>V AND OB>0 THEN M$=O$+" ti!"
510 IF VB>V AND OB=0 THEN M$="Ne radim besmislice!"
520 IF VB<V AND OB>0 AND C(OB)=0 THEN M$="Nemam "+W$
530 IF F(26)=1 AND RM=13 AND VB<>21 AND VB>2 THEN M$="Napadaju sismisi!":GOTO
190
540 IF RM=44 AND RND(2)=1 AND F(24)<>1 THEN F(27)=1
550 IF F(0)=1 THEN LL=LL-1
560 IF LL<1 THEN F(0)=0
570 ON VB GOSUB 610,680,750,750,750,750,750,750,750,1090,1090,1150,1190,1260,1
300,1340,1370,1420,1470,1520,1540,1570,1600,1630,1650,1450,1740
580 IF LL=10 THEN M$="Sveca mi dogoreva!"
590 IF LL=1 THEN LL=0: M$="Sveca je izgorela!":F(0)=0
600 GOTO 190
601
602 REM  naredba 'upomoc'
603


610 PRINT "Znam sledece reci:"
620 FOR I=1 TO V
630 PRINT V$(I);",";
640 NEXT I
650 M$="":PRINT
660 GOSUB 1730
670 RETURN
671
672 REM  naredba 'inventar'
673
680 PRINT "Nosim:"
690 FOR I=1 TO G
700 IF C(I)=1 THEN PRINT O$(I);",";
710 NEXT I
720 M$="":PRINT
730 GOSUB 1730
740 RETURN
741
742 REM  naredbe za kretanje
743
750 D=0
760 IF OB=0 THEN D=VB-3
770 IF OB=19 THEN D=1
780 IF OB=20 THEN D=2
790 IF OB=21 THEN D=3
800 IF OB=22 THEN D=4
810 IF OB=23 THEN D=5
820 IF OB=24 THEN D=6
830 IF RM=20 AND D=5 THEN D=1
840 IF RM=20 AND D=6 THEN D=3
850 IF RM=22 AND D=6 THEN D=2
860 IF RM=22 AND D=5 THEN D=3
870 IF RM=36 AND D=6 THEN D=1
880 IF RM=36 AND D=5 THEN D=2
890 IF F(14)=1 THEN M$="Bup!!! Pao sam sa drveta!":F(14)=0:RETURN
900 IF F(27)=1 AND RM=52 THEN M$="Duhovi mi ne daju da mrdnem!":RETURN
910 IF RM=45 AND C(1)=1 AND F(34)=0 THEN M$="Osecam magicnu barijeru na zapadu
:RETURN
920 IF (RM=26 AND F(0)=0) AND (D=1 OR D=4) THEN M$="Sasvim je mracno":RETURN
930 IF RM=54 AND C(15)<>1 THEN M$="Zaglibio sam se":RETURN
940 IF C(15)=1 AND NOT (RM=53 OR RM=54 OR RM=55 OR RM=47) THEN M$="Sad malo ti
nosi camac!":RETURN
950 IF (RM>26 AND RM<30) AND F(0)=0 THEN M$="Ja u mrak ne idem!": RETURN
960 F(35)=0:RL=LEN(R$(RM))
970 FOR I=1 TO RL
980 U$=MID$(R$(RM),I,1)
990 IF (U$="N" AND D=1 AND F(35)=0) THEN RM=RM-8:F(35)=1
1000 IF (U$="S" AND D=2 AND F(35)=0) THEN RM=RM+8:F(35)=1
1010 IF (U$="W" AND D=3 AND F(35)=0) THEN RM=RM-1:F(35)=1
1020 IF (U$="E" AND D=4 AND F(35)=0) THEN RM=RM+1:F(35)=1
1030 NEXT I
1040 M$="O.K."
1050 IF F(35)=0 THEN M$="Tuda ne mogu!"
1060 IF D<1 THEN M$="Kud' da idem?"
1070 IF RM=41 AND F(23)=1 THEN R$(49)="SW":M$="Vrata su se zalupila":F(23)=0
1080 RETURN
1081
1082 REM  naredba 'uzmi'
1083
1090 IF OB>G THEN M$="Ne mogu da uzmem "+W$:RETURN
1100 IF L(OB)<>RM THEN M$="A gde je?"
1110 IF F(OB)<>0 THEN M$="Koji "+W$+"?"
1120 IF C(OB)=1 THEN M$="Vec ga imam!"
1130 IF OB>0 AND L(OB)=RM AND F(OB)=0 THEN C(OB)=1:L(OB)=65:M$="Imam "+W$
1140 RETURN
1141


1142 REM  naredba 'otvori'
1143
1150 IF RM=43 AND (OB=28 OR OB=29) THEN F(17)=0:M$="Otvorio sam fioku"
1160 IF RM=28 AND OB=25 THEN M$="Zakljucano!"
1170  IF RM=38 AND OB=32 THEN M$="Bas je uzasno!":F(2)=0
1180 RETURN
1181
1182 REM  naredba 'ispitaj'
1183
1190 IF OB=30 THEN F(18)=0:M$="Nesto je tu!!!"
1200 IF OB=31 THEN M$="Odvratno mi je."
1210 IF (OB=28 OR OB=29) THEN M$="Vidim fioku"
1220 IF OB=33 OR OB=5 THEN GOSUB 1260
1230 IF RM=43 AND OB=35 THEN M$="Ima nesto ispod..."
1240 IF OB=32 THEN GOSUB 1150
1250 RETURN
1251
1252 REM  naredba 'procitaj'
1253
1260 IF RM=42 AND OB=33 THEN M$="To su demonske reci!"
1270 IF (OB=3 OR OB=36) AND C(3)=1 AND F(34)=0 THEN M$="Tajanstvena magicna rec
je XZANFAR."
1280 IF C(5)=1 AND OB=5 THEN M$="Napis je na stranom jeziku."
1290 RETURN
1291
1292 REM  naredba 'kazi'
1293
```

```
ka ali bar svetli";F(0)=1
1510 RETURN
1511
1512 REM  naredba 'ugasi'
1513
1520 IF F(0)=1 THEN F(0)=0:M$="Ugasena je."
1530 RETURN
1531
1532 REM  naredba 'pospricaj'
1533
1540 IF OB=26 AND C(16)=1 THEN M$="Hssssss"
1550 IF OB=26 AND C(16)=1 AND F(26)=1 AND RM=13 THEN F(26)=0:M$="PFFT! Gotova s
u!"
1560 RETURN
1561
1562 REM  naredba 'ukljuci'
1563
1570 IF OB=10 AND C(10)=1 AND C(11)=1 THEN M$="Ukljucen je.":F(24)=1
1580 IF F(27)=1 AND F(24)=1 THEN M$="Usisao sam duhove.":F(27)=0
1590 RETURN
1591
1592 REM  naredba 'otvori'
1593
1600 IF RM=43 AND (OB=27 OR OB=28) THEN GOSUB 1150
1610 IF RM=28 AND OB=25 AND F(25)=0 AND C(18)=1 THEN F(25)=1:R$(RM)="SEW":D$(RM
)="Kraj velikih otvorenih vrata":M$="Ljuc se okrece."
1620 RETURN
1621
1622 REM  naredba 'baci'
1623
1630 IF C(OB)=1 THEN C(OB)=0:L(OB)=RM:M$="Izvrseno!"
1640 RETURN
1641
1642 REM  naredba 'skor'
1643
1650 M$=""
1660 S=0
1670 FOR I=1 TO 8
1680 IF C(I)=1 THEN S=S+1
1690 NEXT I
1700 IF S=8 AND C(15)<>1 AND RM<>57 THEN PRINT:PRINT "Prikupio sam sve!":PRINT
```

```
na vrhu grebena
2020 DATA blizu srusenog zida,u sumornoj prolazu kucnoj elektricnoj centrali,
u svecanom hodniku
2030 DATA kraj okovanih drvenih vrata,u svecanoj sali,u podrumu sa zatpranim pr
ozorom,na putu ka litici
2040 DATA kraj ekspera na kome visi kaput,ulaznom holu,u dnevnoj sobi,u tajnom s
obicku
2050 DATA kraj stranih mramornih stepenica,u trpezarijii,u dubokom podrumu s mrtv
ackim sandukom,na putu ka litici
2060 DATA u ormanu,u predvorju,u biblioteci sa zlim knjigama,radnoj sobi sa sto
lom i rupom u zidu
2070 DATA u zlokobnoj sobi punoj paucine,u ledenom prolazu,u avetinjskoj sobi,n
a stremom putu kraj mocvare
2080 DATA na verandi punoj komadica stakla,pred glavnim ulazom,u predjem tornj
u,u vlaznom hodniku
2090 DATA na gornjoj galeriji,na mocvarnoj strani zida,u mocvari,na raskvasenoj
stazi
2100 DATA na krivini,na putu ka gvozdenim vratima,na putu,ispod prednjeg tornja
2110 DATA kod gomile opalog maltera,kod velikog odronjenog kamena,pokraj propal
og svoda,na uglu svete
2120 FOR I=0 TO 63
2130 READ D$(I)
2140 NEXT I
2150 DATA sliku,prsten,magicnu cedulju,pehar,zamotuljak,novcice,kip,svecnjak
2160 DATA sibice,usisivac,baterije,lopatu,sekiru,konopac,camac,superbum,svecu,k
juc
2170 DATA na severu,na jugu,na zapadu,na istoku,gore,dole
2180 DATA vrata,sisalse,avet,fioku,sto,kaput,otpatke
2190 DATA sanduk,knjige,kzanfar,zid,caroliju
2200 FOR I=1 TO W
2210 READ O$(I)
2220 NEXT I
2230 F(18)=1:F(17)=1:F(2)=1:F(26)=1:F(28)=1:F(23)=1:LL=60:RM=57:M$="O.K."
2240 RETURN
2250
2260 REM       Dejan Ristanovic 1984.
2270
```

```
"treba jos da izadjem..."
1710 IF S=8 AND RM=57 THEN PRINT "Dolazak ovde duplira skor";:S=S*2
1720 PRINT "Skor je ";S:IF S>9 THEN PRINT:PRINT "B R A V O. Avantura je zavrsen
a!":END
1730 PRINT:PRINT:INPUT "Pritisni RETURN za nastavak"0$
1740 RETURN
1741
1742 REM  inicijalizacija nizova
1743
1750 DIM R$(63),D$(63),O$(W),V$(V)
1760 DIM C(W),L(G),F(W)
1770 DATA 46,38,35,50,13,18,28,42,10,25,26,4,2,7,47,60,43,32
1780 FOR I=1 TO G
1790 READ L(I)
1800 NEXT I
1810 DATA upomoc,inventar,idi,n,s,w,e,u,d,uzmi,digni,otvori,ispitaj,procitaj,ka
zi,kopaj,upotrebi,popni,upali,ugasi,pospricaj,ukljuci,otkljucaj,baci,skor,sidji
1820 FOR I=1 TO V
1830 READ V$(I)
1840 NEXT I
1850 DATA SE,WE,WE,SWE,WE,WE,SWE,WS
1860 DATA NS,SE,WE,NW,SE,W,NE,NSW
1870 DATA NS,NS,SE,WE,WE,NWUD,SE,WSUD,NS
1880 DATA N,NS,NSE,WE,WE,NSW,NS,NS
1890 DATA S,NSE,NSW,S,NSUD,N,N,NS

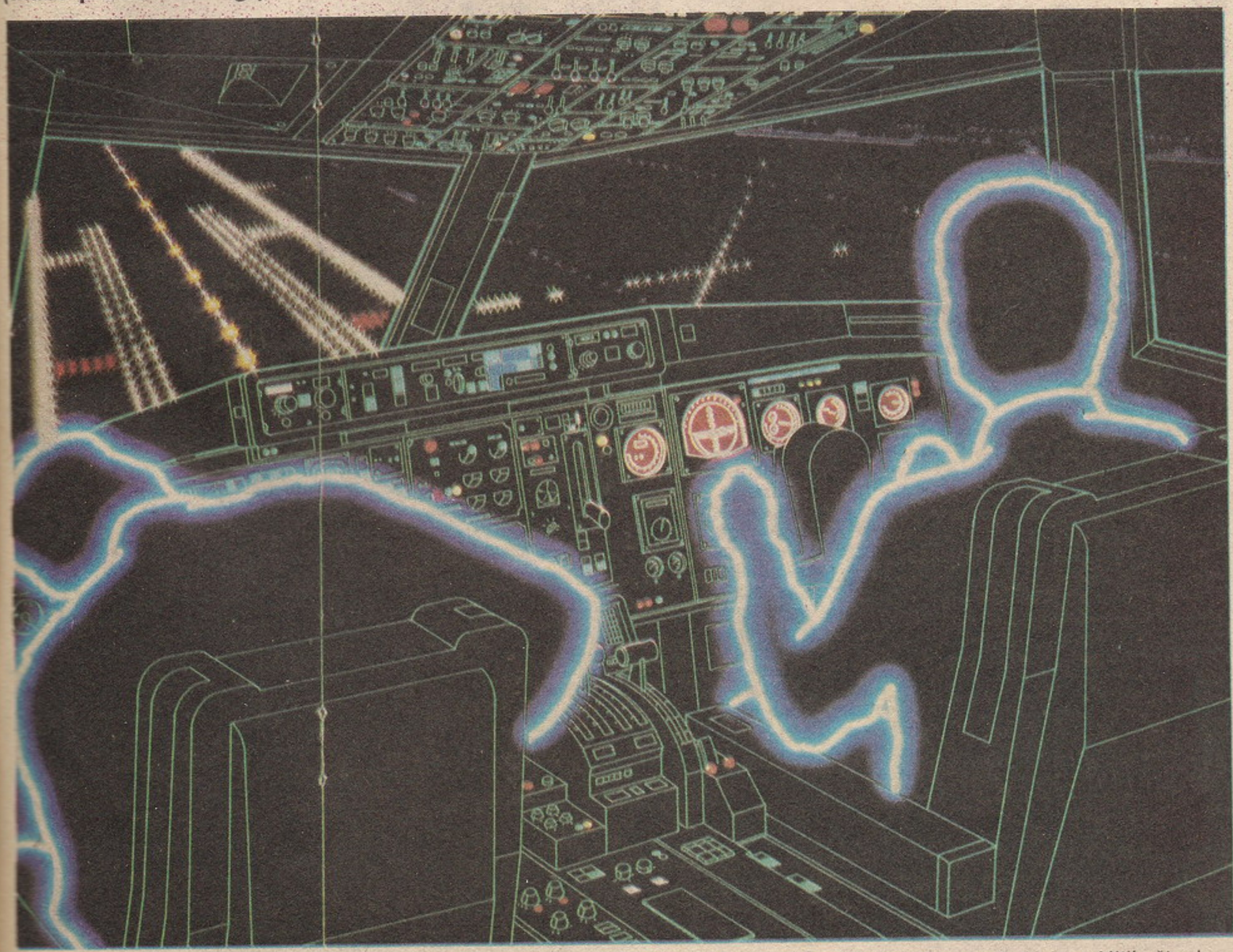
1900 DATA NE,NW,NE,W,NSE,WE,W,NS
1910 DATA SE,NSW,E,WE,NW,S,SW,NW
1920 DATA NE,NWE,WE,WE,WE,NWE,NWE,W
1930 FOR I=0 TO 63
1940 READ R$(I)
1950 NEXT I
1960 DATA u tamnom uglu,u zapustenoj basti,kraj velike gomile drva,u dvoristu k
raj otpadaka
1970 DATA na gradilistu,u sumi,u gustoj sumi,kod suvog drveta
1980 DATA na uglu kuce,na ulazu u kuhinju,u kuhinji,u perionici sudova
1990 DATA u sobi punoj prasine,sa zadnje strane kule,na cistini kraj kuce,na pu
tici
2000 DATA uz zid kuce,iza trema,u tamnom alkovu,u mracnoj sobici
2010 DATA na dnu spiralnih stepenica,u sirokom prolazu,kraj klizavih stepenica,
```

letači bez diplome

Krilata katedra Zorana Modlija

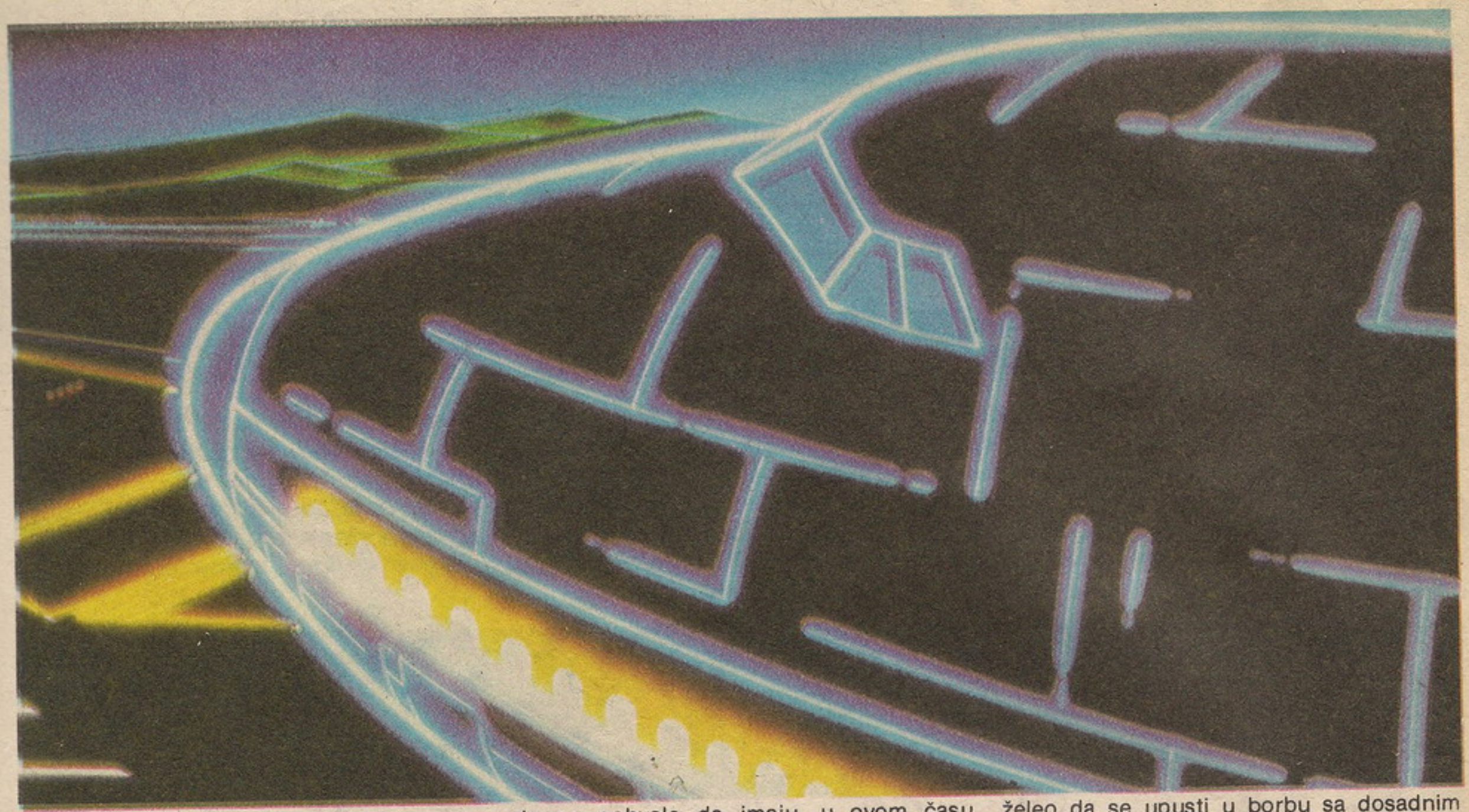
Simulacije letenja

Poseban vid akcionih igara na kućnim računarima obuhvaćen je takozvanim „simulatorima“, koji vam pružaju šansu da se okušate kao vozač najrazličitijih suvozemnih, vodenih, vazdušnih i svemirskih naprava. No, izgleda da su među ovim imitatorima stvarnosti najautoritativniji i najosmišljeniji — mada bez krvi i dima! — simulatori letenja avionom. Sve manje naivni, zahvaljujući opcijama koje nude i letačkom predznanju koje zahtevaju, predstavljaju ozbiljan izazov čak i za profesionalne pilote! „Letenje“ uz pomoć mikroračunara biće, istovremeno, jedno od značajnih poglavlja nove knjige Zorana Modlija (inače profesionalnog pilota i instruktora letenja), „Piste u noći“.



Malo više od igre: Simulacije letenja na Kućnim računarima pružaju pilotima mogućnost da već na zemlji steknu dragoceno letačko znanje i iskustvo

„Filmski glumac Džon Travolta, udobno zavaljen u naslonjaču, prisustvovao je poslovnom sastanku sa izdavačima kompjuterskog softvera, koji su nameravali da iskoriste njegovu popularnost i šarm za propagiranje novih izdanja na tržištu“ — javila je nedavno čitaocima časopisa „Tajm“ novinarka Kristina Garsija (Cristina Garcia). „U tom času neko se setio „Simulatora letenja II“, programa koji svoje korisnike stavlja u kožu Voltera Mitija što deo svog tajnog života provodi kao pilot u kokpitu jednomotornog aviona. Kao pilot amater koji svojim malim mlaznjakom često odlazi sa svog ranča u Santa Barbari, Kalifornija, Travolta nije mogao da odoli, već je uzeo disk i program sa njega uneo u računar. Na zaprepašćenje izdavača, dva sata kasnije Travolta je i dalje bdeo nad kompjuterom, odvažno pilotirajući imagi-



Izazov i za profesionalne pilote: U trenutku kada se na ekranu pojavi duga aerodromska pista i avion uđe u najkritičniju fazu leta, bilo zakuca jače čak i letačkim znalcima

narnim Pajperom 181 Čiroki Arčer, pet hiljada stopa iznad Los Anđelesa!"

Nije samo Travolta zanesen kompjuterizovanim letenjem. Svakog dana hiljade Amerikanaca penju se, zavaljeni u stolice i rukom oslonjenom na džojstik, ka elektronskom plavetnilu Microsoftovog Flight Simulatora, koji žubori kroz elektronska kola IBM Personal Computera, ili Sub Logicovog „Flight Simulatora II", verziji prilagođenoj za „epl", „atari" i „komodor" mašine. Više od 200 hiljada kopija ovih diskova „teških" pedesetak dolara prodato je isto tolikom broju entuzijasta — od onih kojima su jednostavno dosadile konvencionalne video igre, pa do profesionalnih pilota kojima nisu dovoljna uzbuđenja koja im pruža svakodnevni posao. Neki biznismeni redovno lete za Čikago za vreme pauze za prepodnevnu kafu.

„To je najbolja igra koju imam", veli izvesni Karl Hater, sedmogodišnji as iz Karson Sitija, Nevada, koji je jednom leteo simulatorom od Njujorka do Los Anđelesa. „Najteži deo je noćno sletanje u Čikago."

„Pogled iz kabine je neverovatan", oduševljava se Ebot Pejn, pilot iz Orendža, Kalifornija. „Kad se okrećete, okreće se i pejzaž."

Dr Feliks Sasano, hirurg iz Konektikata, zaklinje se da ga je kućni simulator letenja izlečio straha od letenja! Penzionisani pomorski oficir Džon Čartijer tvrdi da se sa 30 sati nedeljno uspešno priprema za predstojeću letačku sezonu.

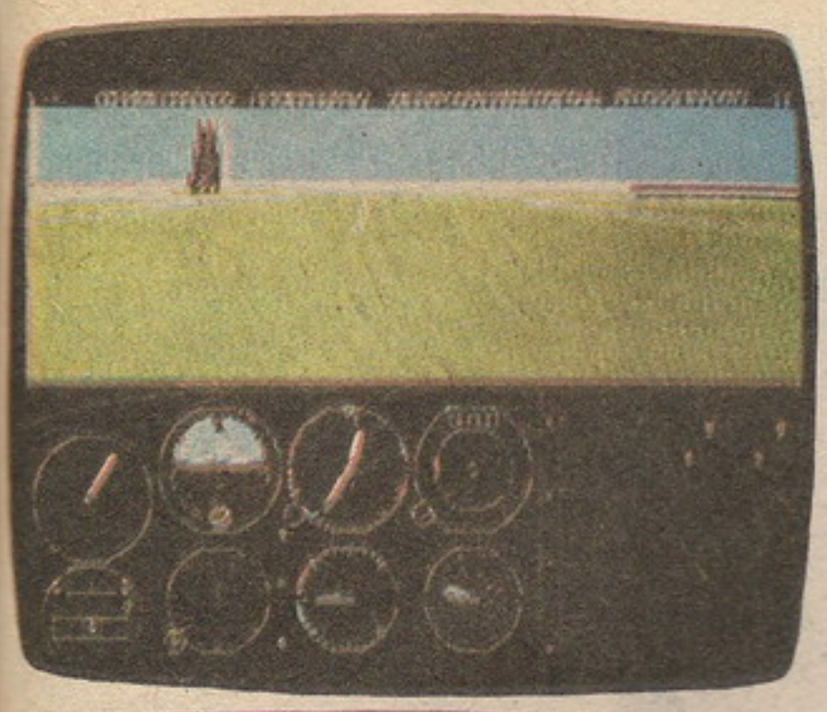
Ovim anegdotama možete da verujete ili ne verujete, ali domaći vlasnici „epla", „atarija" i „Komodora 64" mogu s pravom da se pohvale da imaju, u ovom času najuspešniji, program simulacije letenja za kućne računare koji im dopušta sletanje na ukupno 80 različitih aerodroma. Ključ uspeha ovog programa je u tome što je izuzetna pažnja posvećena detaljima. Koristeći tehniku razvijenu svojevremeno na vojnim uređajima, program daje trodimenzionalni pogled iz ptičje perspektive, gde objekti ravnomerno prolaze ispod aviona, sugestivno menjajući međusobne odnose. Nova verzija ovog programa pokazuje čak i šiljke na kruni Statue slobode u njujorškoj luci dok pilot prolazi kraj nje. Svi brojčanici i merači pravog aviona smešteni su na donjoj polovini ekrana, uključujući i visinomer, veštački horizont, brzinomer i dva kompasa. Tasterima i palicom upravlja se kormilom pravca, gasom, kormilom dubine, krilcima, zakrilcima i mehanizmom za uvlačenje i izvlačenje stajnog trapa. Kompletna radio-navigaciona oprema za noćno letenje je realistična u toj meri da je RAF-ovi piloti koriste za uvežbavanje sletanja. Neki „letači iz fotelje" uznapredovali su od simulatora letenja do prave pilotske škole — ali većina ipak nema nameru da stekne i pilotsku dozvolu!

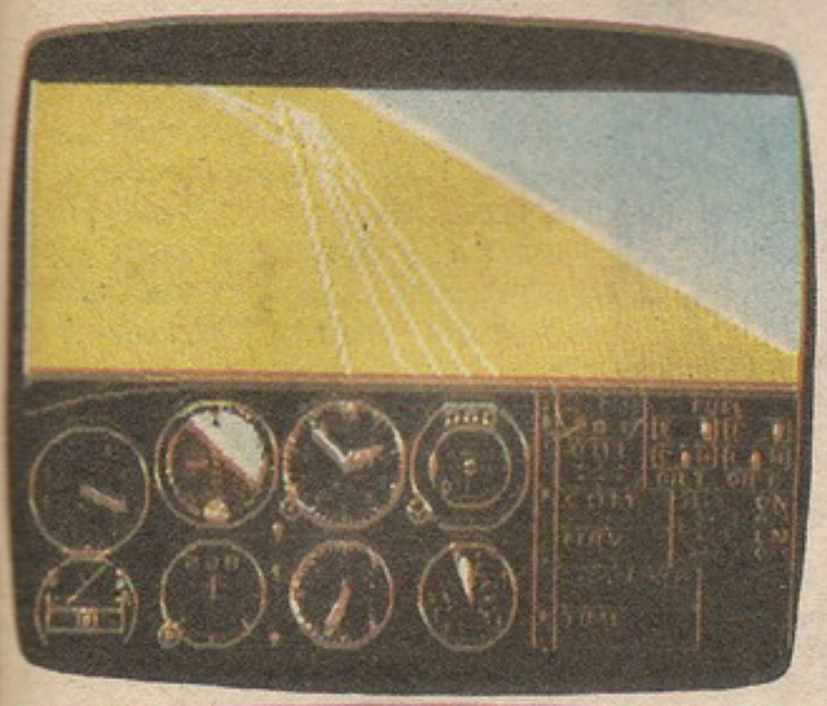
Noć posle teškog dana

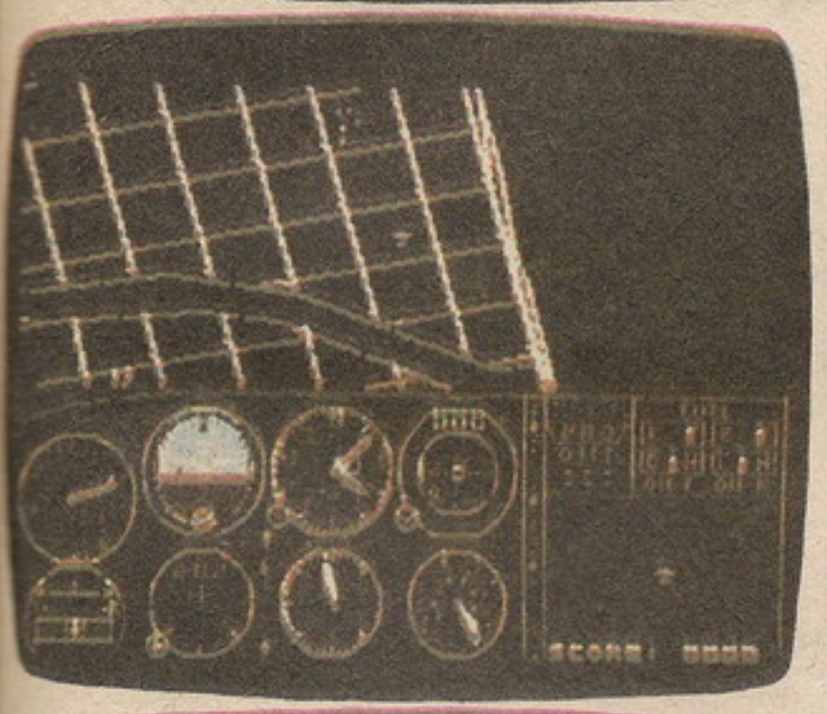
Iz pijeteta prema kućnom mikroračunaru SINCLAIR ZX 81, koji je među prvima, početkom osamdesetih, zakoračio i u naše domove, samo ćemo pomenuti simulator letenja pisan za njega. ZX 81 ustupio je, bez mnogo ljutnje i suza, mesto svojoj mlađoj i jačoj rodbini, ali će se njegovi nekadašnji vlasnici — među njima i autor ovih redova — sa zadovoljstvom sećati uzbudljivih sati sletanja, poletanja i krstarenja koje je nudio Psionov „Flight Simulation", prvi iz kasnije sve bogatije serije programa za letenje avionom u dnevnoj sobi. Takođe, po svim kasnijim merilima, ovaj program nije mogao da se podiči bogzna kakvom animacijom, kvalitetom pružene slike i atraktivnošću izbora. Ali je ono što se odvijalo na ekranu bilo dovoljno pregledno, verodostojno i privlačan izazov za „pilota" koji je želeo da se upusti u borbu sa dosadnim vetrom i gorivom koje polako, ali sigurno nestaje, leteći uz pomoć radio-kompasa ka svom odredištu: jedinoj pisti, što se pružala u pravcu istok-zapad, na sredini ekrana.

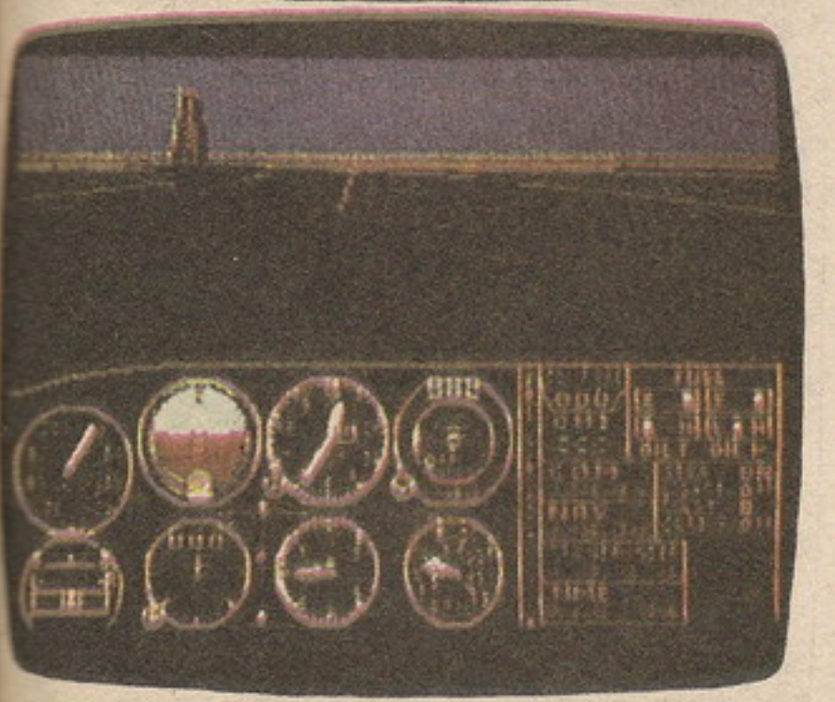
Apsurdno bi bilo redom opisivati sve programe letenja za kućne računare na šarenom svetskom tržištu — iz prostog razloga što svi ovi programi mogu da se, sa manjim odstupanjima, podele na tri grupe: simulacije letenja tromim avionom za uvežbavanje instrumentalnih procedura, simulacije koje traže prefinjenu tehniku pilotiranja i nude akciju i simulacije koje simuliraju — simulator.

„Noćni let" (Nightflite) iz prve grupe, pisan za „spektrum" od 16 K, zaslužuje dužno poštovanje. Prosto je neverovatno da je u tako malo memorije (ne zaboravite da gotovo 8 kilobajta odlazi na podržavanje ekrana) stalo toliko mnogo mogućnosti! Po scenariju ste pilot aviona koji treba da poleti sa ili sleti na pistu što se prostire u pravcu istok-zapad. U izvršenju niza zadataka pomažu vam dva konvencionalna radio-fara (od kojih je jedan smešten u produžetku ose poletno-sletne staze), jedan VOR predajnik, sistem za instrumentalno sletanje ILS i savršeno pregledna i jednostavno dizajnirana instrumentalna tabla u kabini aviona. Pri tom prilazite aerodromu pazeći da ne očešete brda sa severne strane visoka oko 1650 stopa, ili da se ne sudarite sa drugim avionima koji lete na visini preko 2200 stopa. Poletanje, sletanje i održavanje parametara leta ne predstavlja apsolutno nikakav problem, tako da imate dovoljno vremena i mogućnosti da se natenane pozabavite proceduralnim stavkama u letenju: uvežbavate ulaske u krug čekanja (holding) iz raznih sektora, pravite sve vrste proceduralnih zaokreta, presrećete radiotragove zemaljskih radio-navigacionih uređaja, koristite asistenciju autopilota — i na kraju, na par kilometara od praga piste za sletanje, uživate u lepoj iluminaciji aerodroma i okolnih naselja u noći. Po završetku leta, računar na ekranu iscrtava horizontal-









Prizori iz letačkog života: Majkrosoftov simulator letenja omogućuje ljubiteljima računara da na jednomotornoj „cesni 182" lete između 20 aerodroma u SAD; donji deo ekrana je uvek rezervisan za tablu s instrumentima, dok se kroz gornji može uživati u panorami predela

nu i vertikalnu projekciju putanje leta (tek da vidite gde ste sve grešili) i daje ocenu za vašu navigaciju! Verovatno zbog hazarderske žice koja čuči u svakom od nas (malo napetosti nikada nije na odmet), autor programa se pobrinuo da u svakoj od pet mogućnih varijanti letenja tokom „Noćnog leta" sa gorivom budete knap . . . Pistu obavezno dodirujete u trenu kad poslednje kapi benzina ističu iz vašeg rezervoara. Taj problem ćete lako rešiti promenom brojeva na liniji 201, pošto je gotovo ceo program u bejziku.

„Noćni let" je kao poručen za „fiskulturu mozga" kod pilota koji imaju ambicije da se školuju u instrumentalnom letenju, jer pruža mogućnost za beskrajno i bezbolno uvežbavanje takozvane radio-orijentacije u vazduhu, pri čemu treba od niza apstraktnih podataka sa pokazivača u kokpitu sagraditi jasnu i tačnu sliku položaja aviona u prostoru. Jednom rečju, idealno nastavno sredstvo za IFR nastavu u pilotskim školama. Što je još lepše, ovaj — u svojoj klasi najuspešniji — letački program dobio je dostojnog naslednika, program „NIghtflite II", koji je kompletno urađen u mašinskom kodu, dopunjen novim opcijama i, u debrifingu na kraju leta, potpisom autora!

Za razliku od „Noćnog leta", jedan drugi program sličnih performansi, ali ambicioznije zamišljen, „Bac 111", doživeo je na našem tržištu totalni komercijalni krah. Listing ovog programa objavio je časopis „Your Computer" krajem pretprošle godine, pompezno ga najavljujući kao „najrealističniji simulator ikada publikovan za kućne računare, koji uspešno zamenjuje pilotske trenažere od nekoliko miliona funti!" Naše vredne hakere nije mrzelo da provere ovaj hvalospev, pa su „Bac 111" ukucali u svoje „spektrume". Naravno, ispostavilo se da je pravi trenažer bolji od ovog „spektrumovog" bar onoliko puta koliko je vaš računar jeftiniji od putničkog aviona BAC 111! Dobro, to se moglo i očekivati. Međutim, sâm program doživljavao je hronične traume: imao je običaj da na ekranu, umesto urednih brojki na letačkim instrumentima, ispisuje obične gluposti, nakon leta (koji nikada nije mogao da bude krunisan sletanjem jer je na milju i po od piste obavezno zaribao, ispisujući ono beznadežno „Integer out of range") ekran sa slikom instrumentalne table zauvek se brisao, itd. Neke mane su se dale ispraviti manjim intervencijama, ali entuzijazam oduševljenih pilota bio je, na žalost, veći od njihovog hakerskog znanja.

Zanimljivo je da „Bac 111" daje dosta privlačnu mogućnost da letite po veoma verno datoj karti Glazgova i Edinburga, zamišljajući da iza vas sedi stotinak putnika koji se voze brzinom od oko 700 kilometara na čas. Kao u pravom putničkom mlaznom avionu. Ukoliko vas interesuje konačna sudbina ovog programa (možda je u međuvremenu ipak postao savršen!), možete da se obratite njegovom autoru. To je Rod Hopkins, 116 Pilmuir Street, Dunfermline, Fife, Scotland.

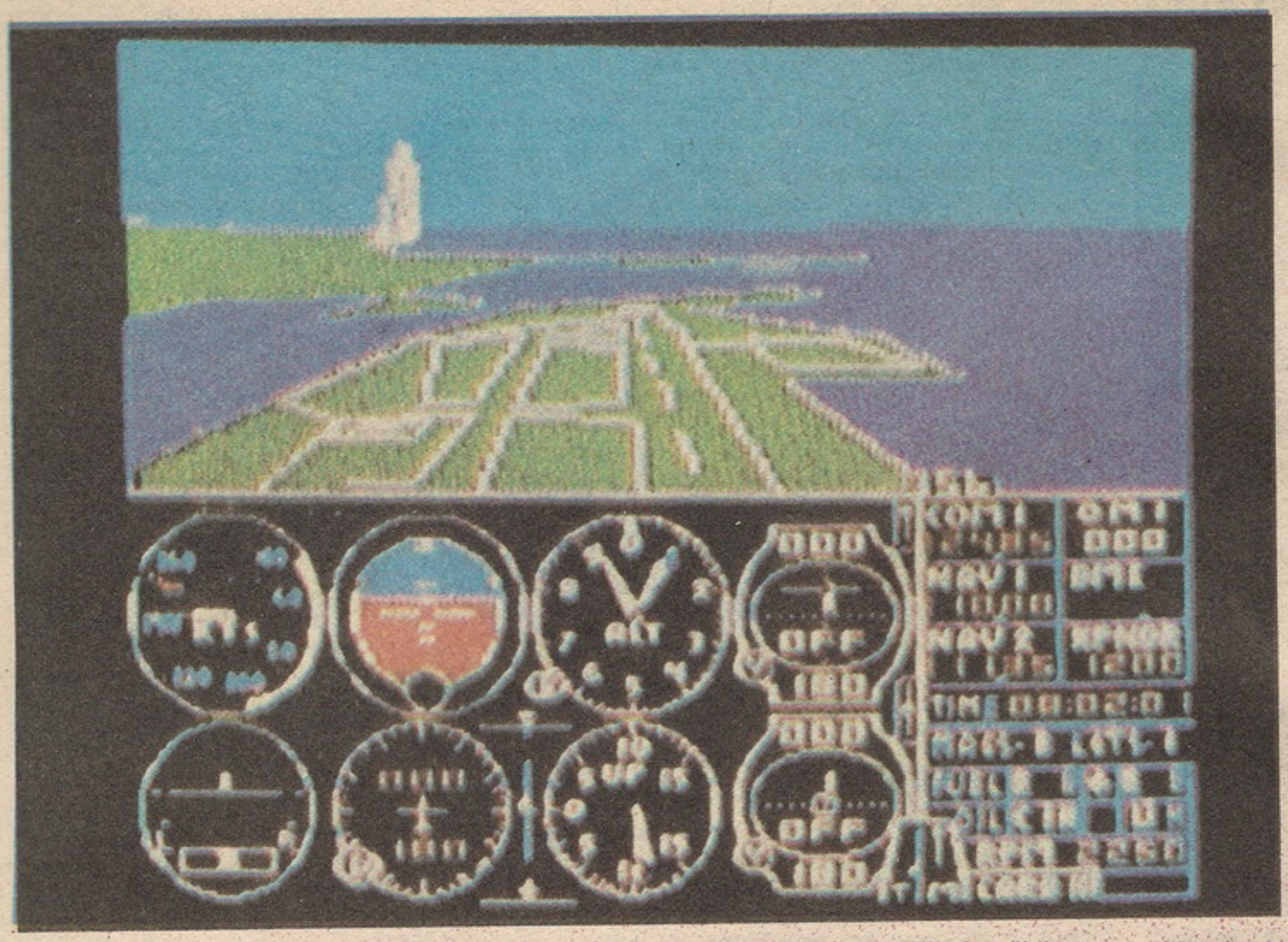
„Fantom" u akciji

Vlasnici „spektruma" u ovom trenutku proglašavaju za letački program godine simulaciju letenja pod nazivom „Borbeni pilot" (Fighter pilot). Program ni najmanje nije naivan. Sedite za komandama „fantoma" F—14 TOMCAT, lovačkog mlaznjaka koji je i za skormne kućne sulove opremljen veoma naprednom elektronikom: uz standardne pilotažne instrumente i indikatore položaja zakrilaca i stajnih organa, imate osmatrački radar koji pokazuje položaj neprijateljskih aviona u odnosu na vaš, naznačujući njegovu tačnu udaljenost i kurs u kojem treba leteti da biste ga iznenadili. Poseban „kompjuter" proračunava visinu leta cilja i vodi projektil koji ispaljujete ka njemu. „Fantom" može da ispali šest projektila „feniks" na šest odvojenih ciljeva u isto vreme. Svaki projektil je kompjuterski vođen ka svom cilju. Pri tom imate mogućnost i za pravi „dog fajt", ukoliko se odlučite za borbu prsa u prsa sa protivnikom. Ali, morate da budete oprezni: dok se vi na par desetina kilometara visine jurite sa jednim od neprijateljskih lovaca, ostali će se kao osice obrušiti na vaš aerodrom, uništiti ga i tako vam definitivno onemogućiti povratak u bazu nakon stratosferskog leta. Ako vas u plemenitom naporu da očistite nego od drskih uljeza zatekne mrak, imate šansu da izvedete napeto noćno sletanje. Pretpostavimo da se niste (opet!) razbili o pistu, zaboravili da izvučete točkove ili oduzmete gas, već bez greške ponovo došli u naručje majčici zemlji; tada će rezerve municije i goriva biti dopunjene, a vama biti dozvoljeno da ponovo krenete na zadatak.

Letenje „fantomom" je prava poslastica. Sposoban je za sve manevre, uključujući i komplet svih mogućih akrobacija. Na komande reaguje trenutno i lako, a kad mu dodate „forsaž" — garantovano probija zvučni zid! Izuzetno je zabavan, jer predstavlja prvi pravi akcioni simulator letenja. Mala napomena za one koji tek treba da nabave ovaj program: prilikom unošenja programa u računar obavezno ukucajte CLEAR 65535: LOAD "" CODE (ukoliko onaj od koga ste program dobili to već nije uneo u program kao gotovu bejzik naredbu); sa tako, do maksimuma podignutim ram-topom, „Fighter pilot" neće blokirati prilikom startovanja igre.

Još jedan program za računare „spektrum" (one sa 48 K rama) možemo da uvrstimo u simulacije letenja koje pružaju nešto malo temperamentniju akciju: to je PSIONOV „Let" (Flight), kojim se ova firma izuzetno ponosi. Ne bez razloga. Kroz vetrobransko staklo pilotske kabine vidi se skroman i stilizovan pejzaž, ispresecan sa nekoliko plavih jezera, dovoljno duga pista glavnog i osetno kraća pista sportskog aerodroma. Jednomotorni avion je prilično nestašan, zbog vrlo osetljive aerodinamike. Promena jedne komande obično izaziva niz drugih efekata. Ako uvedete avion u nagib, promena u opstrujavanju vazduha prouzrokuje njegovo skretanje sa dotadašnjeg pravca — ali i gubitak visine, pa morate hitro da se pomognete gasom, povećavajući obrtaje motora. Kod ponovnog povratka na pravolinijski let, malopređašnji višak obrtaja izazvaće neželjeno penjanje aviona, pa ste ponovo prinuđeni da intervenišete i komandom gasa, i tako dalje. Letelica je osetljiva na turbulenciju, a efekti vetra koje vam računar nudi (pod uslovom da ih vi prihvatite) čine navigaciju i završni čin sletanja još težim. Sa raspoloživim gorivom i ekonomičnom brzinom krstarenja možete da letite punih 40 minuta (što bi odgovaralo letu od Beograda do Sarajeva, na primer).

Zanimljivo je da su autori ovog programa pedantno vodili računa i o praktičnom plafonu leta aviona, pa tako brzina penjanja i snaga motora ravnomerno opadaju sa visinom, da bi, sa dostizanjem plafona, svaki dalji pokušaj penjanja ostao uzaludan i — ukoliko ste i dalje tvrdoglavi — rezulti-



Programersko remek-delo: „Simulator letenja II" firme SubLogic obezbeđuje letačima na „eplu", „atariju" i „komodoru 64" izuzetno veran doživljaj pilotiranja

rao padom aviona u kovit. Sletanje je izuzetno složeno i zahteva savršenu koordinaciju svih komandi (uključujući i pedale!). Ako uspete da živi i zdravi liznete pistu, smatrajte to velikim uspehom i ne osvrćite se na eventualne zlobne komentare računara, kao što su „ne skakućite na sletanju!", „suviše tvrdo sletanje", itd.

Pošto se radi o lakom sportskom vazduhoplovu, pucnjava nije predviđena, ali zato i u „Letu" možete da radite akrobacije do mile volje. Problem je u dosta inertnoj komandi dubine, pa je radijus petlje ogroman; ako ne mislite da se razbijete o zemlju, ne krećite u vertikalne evolucije na visini manjoj od nekoliko hiljada stopa!

Od radionavigacione opreme, „Let" vam pruža radio-kompas, DME (uređaj za merenje udaljenosti od radio-sredstva) i ILS (sistem za instrumentalno sletanje). Zbog nestabilnog aviona, navigacione procedure su prilično neugodne, jer je letač prevashodno okupiran tehnikom letenja. No, možda je baš zato i izazov veći.

Krilati „komodor"

Poznate prednosti „Komodora 64" nude još atraktivnije vizuelne doživljaje. Time prvenstveno mislimo na početku teksta pomenuti program „Flight Simulator II". Ali, ako želite da se okušate kao pilot borbenog aviona „herijer", koji je inače jedan od retkih uspelih i masovno proizvođenih aviona za vertikalno poletanje i sletanje, predlažemo vam da ukucate program „Flight 64", čiji je listing objavljen u ovogodišnjem februarskom broju časopisa „Your Computer". Uslov je da imate dovoljno (pilotskog!) strpljenja i — „Komodor 64", naravno . . . Možda će vam nešto značiti podatak da, zahvaljujući programu „Flight 64", slećete na napuljski aerodrom.

Jedan od najsuperiornijih letačkih programa do sada načinjenih za kućne računare svakako je „Solo Flight". Ovogodišnje je produkcije, a autor Sid Meier namenio ga je računarima „atari", IBM PC i „Komodor 64". Na primeru ovog programa obradićemo principe upravljanja avionom, što će vam, sigurni smo — uz manje dodatke (ili oduzetke!) — koristiti u letenju kućnim simulatorima sličnog ranga.

„Solo Flight" raspaljuje maštu onih koji bi želeli da upoznaju izazove pilotiranja vlastitim malim avionom. Na ekranu je prilično verna trodimenzionalna slika okolnih predela i kompletna letačka instrumentacija. U ulozi pilota poštanskog aviona treba, u trci s vremenom, mehaničkim kvarovima i meteorološkim neprilikama, da raznesete poštanske pošiljke svojim krhkim jednomotorcem prema propisanom redu letenja.

IZBOR: jednim od funkcionalnih tastera na računaru birate igru za vežbanje (FLYING) ili igru za iskusne pilote (MAIL PILOT), zatim jednu od ponuđene tri države Amerike (Vašington, Kanzas ili Kolorado) u kojoj želite da letite, težinu nivoa igre, uslove letenja — CLEAR (mirno i čisto vreme, pogodno za školske krugove na lokalnom aerodromu), WINDY (uslovi s vetrom), LANDING (gde se avion postavlja u kratak final za sletanje) i IFR (instrumentalni let u oblacima sa niskom bazom) — da biste, konačno, sebi dodelili i željenu ulogu u predstojećoj avanturi: pilota-učenika (STUDENT), pilota sa dozvolom (PRIVATE pilot), iskusnog pilota (SENIOR) ili kapetana vazduhoplova (COMMAND).

OSNOVNI PRINCIPI LETENJA: najelementarnije letenje podrazumeva upotrebu dve komande — palice i ručice gasa. Palicom (joystick-om) menjate visinu, a koamndom gasa brzinu leta aviona. Palicom zaokrećite, penjete ili ponirete. Kad palicu nagnete udesno, i avion se naginje u istu stranu, skrećući udesno. Primetićete da avion ostaje u tom položaju iako ste palicu vratili u centar. Avion ćete „izravnati" odgovarajućim pritiskom palice u suprotnu stranu. Guranje palice unapred (prema ekranu) izaziva obaranje nosa: vaš avion ponire. Povlačenjem palice diže nos. Snagu vašeg motora kontrolišete komandom gasa. Preporučljivo je da maksimalnu snagu koristite tokom poletanja i penjanja, nešto manju (ekonomičnu snagu) za krstarenje, dok je minimum snage potreban u prilazu za sletanje. Postoji uzajamna veza između ugla nosa aviona u odnosu na horizont (takozvani PITCH) i procenta snage potrebne da se održi horizontalni let: male brzine zahtevaju veći napadni ugao — i obratno. Mnogi piloti malih aviona napadnim uglom kontrolišu brzinu, a snagom motora visinu leta. U svakom slučaju, bilo kakva promena jednog elementa utiče na ostale parametre leta. Da postignete savršenu kontrolu nad avionom tokom finalnog prilaza, treba usaglasiti brzinu, snagu, ugao nosa aviona i vertikalnu brzinu propadanja tako, da održite nepromenjenu vazdušnu brzinu (airspeed) i vrednost propadanja (descent rate) sve do tačke dodira piste. U program „Solo Flight" ugrađene su sve ove karakteristike — a time i problemi koje treba savladati kako biste leteli i sletali pod svim okolnostima.

POGLED IZ KABINE: gornja polovina ekrana prikazuje vaš avion gledan s repa i teren ispred njega. Ukoliko letite na maloj visini, uočićete i senku aviona na zemlji! Aerodromi i njihove piste oivičeni su crnom, planine i antene VOR predajnika belom, a farme sivom bojom. Prilikom leta kroz oblake ili iznad njih, objekti na zemlji nastaju. Iako vam osnovna vizura pruža pogled kroz vetrobransko staklo aviona, odgovarajućim kurzorima možete da se osvrnete i okolo i iza sebe.

INSTRUMENTI: donja polovina ekrana sadrži letačku instrumentaciju: **visinomer** (kada se mala kazaljka pomeri za jedan podeok, a velika za pun krug, popeli ste se na hiljadu stopa), **brzinomer** (pokazuje brzinu leta od 0 do 180 čvorova), **veštački aviohorizont, indikator snage motora,** upo-

zoravajuće svetlo za **temperaturu motora, merač goriva,** svetiljke upozorenja za **stajni trap** i **kočnice,** pokazivač sistema za instrumentalno sletanje **ILS** (pomaže pilotu da utvrdi da li je u prilazu za sletanje ispod, iznad ili na propisanoj ravni poniranja), dva **VOR indikatora,** koji pokazuju magnetski smer od zemaljskih VOR stanica ka avionu (tzv. „radijale"). Tu su još i četiri brojčana pokazivača, veoma važna: prvi pokazuje vrednost pozitivnog ili negativnog napadnog ugla nosa aviona (PITCH), drugi obaveštava pilota o stepenima izvučenih zakrilaca (FLAPS), treći označava kompasni kurs leta aviona (HEADING), dok četvrti meri vertikalnu brzinu poniranja ili penjanja u stopama u minutu (VARIO).

OPASNOSTI: ako vam đavo ne da mira, pa izaberete teže nivoe igre, oprema vašeg aviona postaje sklona otkazima. Treperenje sijalice za temperaturu motora signal je da se ovaj pregrejao i da će vrlo brzo biti izbačen iz stroja. Zato, što pre pronađite mesto za sletanje. Visinomer, brzinomer, pokazivač kursa i VOR indikatori takođe mogu da otkažu, očitavajući nulu. Slični problemi mogu da snađu i veštački horizont. Sletanje na bilo koji aerodrom preporodiće avion.

Setite se tri osnovna pravila letenja u vanrednim (emergency) situacijama:

1. Zadržite kontrolu nad avionom.
2. Proučite situaciju i preduzmite odgovarajuću akciju.
3. Sletite čim vam okolnosti dozvole.

Većina vanrednih događaja u avionu ne zahteva trenutno reagovanje (izuzetak je otkaz motora u poletanju). Pilot obično ima vremena da pažljivo isplanira odluku. Neupotrebljivi pilotažni i motorski instrumenti mogu donekle da se kompenzuju pokazivanjima preostalih. U slučaju otkaza motora treba zadržati bezbednu visinu (palicom održavati minimalnu sigurnosnu brzinu!) i, tokom ovog „jedrenja", proračunati sletanje na alternativni aerodrom duž rute. „Solo Flight" avion ima finesu 9 prema 1, što mu, drugim rečima, omogućava da sa visine od hiljadu metara preleti devet puta veće rastojanje. Višekratno aktiviranje „E" (emergency) tastera provešće program kroz sve moguće opasnosti predviđene ovom simulacijom.

Tekuće meteorološke prilike prikazane su na dnu ekrana: pravac i brzina vetra, baza oblaka u stopama i vidljivost u miljama. Sletanje u uslovima sa vetrom postaje neprijatno, posebno kad duva bočno na pistu. Upotrebite minimalno izvučena zakrilca, povećanu vazdušnu brzinu i klizanje aviona u stranu vetra (ukrštanjem komandi pravca i nagiba) kako biste kompenzovali njegov uticaj. Niski oblaci često zahtevaju prelazak na instrumentalno letenje, mada možete da izaberete i let iznad oblaka.

SLETANJE: sletanje je verovatno najteža operacija u letu kojom treba da zagospodarite. Dok prilazite jednom od dvadesetak mogućih aerodroma, gledajte da što pre poravnate avion sa pravcem odabrane piste. Kada „oživi" ILS pokazivač, lakim penjanjem ili poniranjem centrirajte se na njegovom displeju. Izvucite stajni trap, smanjite gas i podesite nos aviona tako da održite propadanje od 400 ili 500 stopa u minutu. Tik pred pistom smanjujte vrednost poniranja laganim podizanjem nosa aviona. Kontrola brzine je krajnje kritična. Suviše brzine — i avion će preplivati celu stazu ili se neće zaustaviti na raspoloživom prostoru. Premala brzina dovešće do sloma uzgona tokom ravnanja i sletanja! Pošto je avion dodirnuo pistu, oduzmite potpuno gas, aktivirajte kočnice i potrudite se da stanete pre kraja staze. Na nekim kraćim pistama to ćete teško izvesti ukoliko ne praktikujete takozvani „short field approach" (zakrilca izvučena do kraja, minimalna dopustiva brzina), sa sletanjem veoma blizu praga.

INSTRUMENTALNO LETENJE: instrumentalno letenje je najsloženiji vid letenja, u kojem pilot, pored dužnosti koje važe i za VFR letenje (poletanja, krstarenja i sletanja), mora dobro da se snalazi i u uslovima lošeg vremena i loše vidljivosti, bez mogućnosti da položaj aviona određuje uz pomoć neba i zemlje, a orijentaciju da održava prema objektima koje osmatra kroz prozore kabine. Dobro instrumentalno letenje podrazumeva unakrsno proveravanje instrumenata, lako opažanje, dobru navigacionu pripremu i sposobnost brzog prihvatanja, proučavanja i reagovanja na podatke dobijene sa instrumenata. „Solo Flight" u instrumentalne postupke uključuje navigaciju po VOR-u, tehniku vođenja aviona po ILS-u i instrumentalne prilaze na velikoj i maloj visini. Pilot mora da ih razume i uspešno izvodi — ukoliko misli da „preživi" izazove instrumentalnog letenja.

VOR NAVIGACIJA: zasniva se na nizu zemaljskih radio-predajnika (VOR) koji isijavaju radio-signale. Ove signale primaju instrumenti u pilotskoj kabini, prevode ih i očitavaju kao smerove od određene VOR stanice ka avionu. Određivanje tačne pozicije aviona zasniva se na merenju uglova između radijala (radio-smerova od VOR stanice prema avionu), ili na praćenju jednog od radijala, pri čemu se štopericom meri vreme leta od njihovog izvora (dakle, poznate VOR stanice). Većina zemalja opremljena je kompletnim VOR sistemima, u kojima bilo koja tačka maršrute može da se odredi u preseku radijala sa više VOR stanica. U „Solo Flightu" je svaki aerodrom definisan tačkom preseka dva radijala, čija su izvorišta dva raspoloživa VOR predajnika. Dodajmo da je vrednost ovakvih radijala, u stvari, izražena stepenima kruga od 0 do 360; ukoliko se avion okrene i zaplovi u kursu za 180 stepeni različitom od vrednosti očitanog radijala, linija leta odvešće ga tačno ka VOR stanici. Severno od VOR stanice proteže se radijal 360, istočno je 090, južno 180, a zapadno 270 (razume se, sa svim ostalim međuvrednostima). Da pronađe željeni aerodrom u „Solo Flight-u", pilot mora da odredi presek dva određena radijala (radial intersection) sa oba VOR-a. Kako? Jednostavno treba da presretne jedan od zadatih radijala i po njemu leti sve dok se ne susretne sa traženim radijalom druge VOR stanice. Na primer, na mapi Kanzasa, aerodrom Kanzas Siti nalazi se na radijalu 036 VOR-a 2, sa pragom glavne piste u tački preseka sa radijalom 067 VOR-a 1. Da bi pronašao aerodrom u uslovima loše vidljivosti, instrumentalni pilot će zauzeti kurs 036° po radijalu 036 VOR-a 2 i ponirati do visine neuspelog prilaza (200 do 500 stopa iznad nivoa aerodroma), sve dok ne preseče radijal 067 VOR-a 1.

SETITE SE: dobar instrumentalni pilot obično će izvesti samo jedan prilaz ka pisti na koju sleće. Ako pilot izvede više neuspelih prilaza, količina adrenalina u krvi naglo raste, gorivo neumitno otiče iz avionskih rezervoara, a do najbližeg alternativnog aerodroma treba leteti još koju stotinu kilometara — i to po nesnošljivom vremenu! Najviše što pilot može da učini jeste da prvi prilaz uradi valjano — sa pravom brzinom, na pravom visini i avionom u konfiguraciji za sletanje — i da bude pripravan za sletanje, ukoliko pistu ugleda tek negde oko tačke neuspelog prilaza!

Sobni džambo-džet

Većina letačkih programa za kućne računare pokušava da vas ubedi da ste bogomdani pilot aviona. Ali, u stvarnosti, nakon završene teoretske obuke u pravoj vazduhoplovnoj školi (a mnogo pre no što će sesti u jednomotorni ili dvomotorni avion za školovanje u instrumentalnom letenju) pilot je obavezan da savlada program obuke na elektronskom simulatoru. IFR obuka na simulatoru identična je onoj u avionu — ali mnogo ekonomičnija i bezbednija. Sa početnih vežbi u osnovama instrumentalnog letenja, preko orijentacije u prostoru i vođenja aviona pomoću ADF-a (radio-kompasa), VOR-a i drugih radio-navigacionih sistema, stiže se do proceduralnog letenja, prilaza i sletanja po ILS-u. Jer, stara pedagoška krilatica svih nastavnika letenja veli:

„Ono što na zemlji znaš za peticu, u vazduhu ćeš jedva znati za trojku!"

Zato su vazduhoplovstvom zadojeni programeri smislili još jedan „trik": simulacije letenja koje na kućnim računarima simuliraju — simulatore! Drugim rečima, prvo savladajte simulator simulatora, a zatim možete i zaista da se vinete u vazduh na simulatoru aviona . . .

Da vas razuverimo: nije reč ni o kakvom pleonazmu i prodaji iste robe u drugom pakovanju. Suštinska razlika između „kućnih aviona" i „kućnih simulatora" postoji. Umesto da dragocenu memoriju troši na manje ili više uspešne vizure raznoraznih pejzaža, građevina i pista sa prilaznim svetlima, računar odvaja ogroman prostor za smeštanje novih podataka. Simulator simulatora nema „prozore" kroz koje možete da osmatrate nego i zemlju, niti grafiku visoke rezulucije. Ali zato daje sofisticirane proračune ogromnog broja drugih elemenata leta, s kojima bi se rado uhvatio u koštac čak i rasni i iskusni kapetan dugolinijske vazdušne plovidbe. Takav je, između ostalih, i program Engleza Dejvida Riza (David Rees), „Flight Simulator 747", koji vas smešta u trenažer džampo-džeta „boing 747" i omogućava prekookeanski let iz Amerike za Evropu — i obratno — po svim principima long-range navigacije, uz mogućnost da povežete aerodrome u Njujorku, Glazgovu, Edinburgu, Mančesteru, Birmingemu, Londonu („Hitrou", „Getvik" i „Stansted") i Parizu („Orli"). Program je, inače, rađen u naprednom bejziku za „Komodor 64", ali je upotrebljiv i na drugim mikro računarima uz odgovarajuća prilagođenja.

Na pitanje: da li simulacije letenja na kućnim računarima pružaju i nešto više od igre? — odgovor je nedvosmislen: da. Pre samo nekoliko desetina godina, skromni link-treneri po vojnim i kompanijskim školama štedeli su tone i tone goriva (i isto toliko novca), pružajući pilotima mogućnost da na zemlji steknu dragoceno letačko znanje i iskustvo.

Danas i najobičniji kućni računar uveliko nadmašuje nekadašnje „čegrtaljke".

Eto male osvete starog Voltera Mitija.

Zoran Modli

programator EPROM-a

Računari u domaćoj radionosti

Hardverska koncepcija programatora EPROM-a, koju smo opisali u prošlom broju, omogućava potpunu programsku kontrolu svake funkcije prilikom čitanja, testiranja, programiranja i verifikacije EPROM-a i, samim tim, ostavlja korisniku potpuno odrešene ruke da odredi funkcionalnu koncepciju i prilagodi uređaj svojim viđenjima i svojim — potrebama. Prva verzija programa za rad uređaja bila je napisana u bejziku. Iako je bio urađen tako da se što brže izvršava, u tzv. „linijskom kodu", bez ijedne GO SUB naredbe, čitava procedura programiranja, od kopiranja master EPROM-a do verifikacije isprogramiranog EPROM-a, trajala je za čip 2732 (ROM računara „galaksija") petnaestak minuta — gotovo četiri puta duže nego što je moguće! To je, psihički, zamorno čak i ako se uređaj koristi samo jedanput mesečno. Bejzik, uz to, nije dobar iz još jednog razloga: računar „galaksija", u osnovnoj verziji, ne poznaje naredbe IN i OUT. One su ugrađene tek u ROM 2. One se, doduše, mogu zameniti naredbama tipa BYTE, ali u nešto drugačijoj hardverskoj konfiguraciji. Ni mašinski jezik, na žalost, nije najsrećnije rešenje: zbog žive komunikacije računara i korisnika u toku rada uređaja, program bi morao da bude veoma glomazan. Zato smo se, kad je reč o Spectrumu, odlučili za „solomonsko" rešenje koje koristi prednosti oba programska jezika: komunikacija, kod koje brzina nije bitna, urađena je u bejziku, a sve izvršne rutine u mašinskom jeziku.

Naš programator EPROM-a ima četiri osnovne funkcije — kopiranje master EPROM-a u memoriju, testiranje EPROM-a koji treba isprogramirati, programiranje i provera da li je programiranje bilo uspešno i dve pomoćne: izračunavanje zbira svih bajtova (tzv. „ček sum") i listanje sadržaja EPROM-A. Pošto način na koji su rešene ove funkcije zadire u samu koncepciju programatora, objasnićemo kriterijume za svaku od njih i dati primer programskog rešenja u bejziku. Primeri su urađeni za EPROM 2732, ali važe, uz promenu parametara, i za sve ostale tipove sa kojima može da radi opisani uređaj.

Kopiranje

Za razliku od profesionalnih mašina za programiranje, naš programtor ne može da vrši direktno kopiranje EPROM-a u EPROM. Zašto? Još jedno skupoceno podnožje sa kočnicom... On prihvata samo program koji se nalazi u RAM-u (ili u ROM-u!) računara. To, praktično, znači da je master EPROM potrebno najpre prebaciti u RAM. Adresa od koje će sadržaj EPROM-a biti smešten nije bitna — važno je da ima dovoljno prostora za čitav program. „Galaksija" sa 6 K RAM-a, i Spectrum sa 16 K mogu, zato, da programiraju samo tipove 2716 (2 K) i 2732 (4 K)!

```
10 REM KOPIRANJE EPROMA U RAM
20 OUT 127, 144 — aktiviraj PIO i postavi portove
30 OUT 93, 132: OUT 93, 4 — resetuj programski brojač
40 OUT 63, 80 — postavi EPROM u mirno stanje (stand by)
50 OUT 63, 144 — postavi EPROM u modalitet „čitanje"
60 FOR N=24000 TO 24000+4095 — počni od adrese 24000, a završi sa adresom 28095
70 LET K=IN 31: POKE N,.K — pročitaj podatak sa porta A i upiši ono što si pročitao na adresu N
80 OUT 93,5: OUT 93, 4 — povećaj programski brojač za 1 — pređi na sledeću adresu EPROM-a
90 NEXT N — pređi na sledeću adresu u RAM-u i nastavi da čitaš sve dok ne dođeš do adrese 28095
100 OUT 63, 80 — vrati EPROM u neutralno stanje
110 PRINT „KOPIRANJE GOTOVO"
```

Postupak čitanja ostalih tipova EPROM-a je istovetan — samo su podaci u OUT instrukcijama i, naravno, kapacitet (broj bajtova koje treba pročitati) drugačiji. Početna adresa (24000) i kapacitet EPROM-a (4095) se unose preko naredbe INPUT, dok se svi ostali podaci kreiraju programski na osnovu podatka o tipu EPROM-a.

Nakon procedure kopiranja, program, u finalnoj verziji, automatski prelazi na rutinu za proveru „ček sum" kopije u RAM-u. Ako se zbir bajtova slaže sa zbirom bajtova u master EPROM-u, kopiranje je bilo uspešno i može se preći na proceduru programiranja. Ovaj povratak, u konačnoj verziji, nije automatski. Program se vraća na „meni" i zaustavlja dok ne dobije nova uputstva. Ako „ček sum" master EPROM-a nije poznat, korisno je pre programiranja izlistati kopiju u RAM-u. Ovakvo rešenje, povrh svega, omogućuje modifikacije i dorade sadržaja EPROM-a pre programiranja.

Testiranje

Pre programiranja treba obavezno proveriti da li je EPROM u koji treba uneti sadržaj prazan ili nije. Ovo testiranje se obično postavlja tako da bude bezuslovno — ako EPROM nije prazan, program se vraća na „meni" i zahteva od korisnika da u podnožje postavi novi EPROM. Ovakvo rešenje izgleda jedino logično sve dok ne dođemo u situaciju da neki EPROM doprogramiramo — EPROM koji nije „napunjen do vrha" može se „dopuniti" kad god se poželi. Naravno, ako to dozvoljava program koji upravlja programatorom. Objavljeni program, na žalost, ne dopušta takvu mogućnost, ali se ona može ugraditi veoma jednostavno.

U praznom EPROM-u u svaku memorijsku lokaciju je „upisan" broj 255. Program za testiranje je istovetan sa programom za kopiranje u RAM osim u programskoj liniji 70. Umesto da ono što je pročitao upiše u RAM, program proverava da li je to broj 255 i, ako nije, vraća se na „meni" po nova uputstva. Linije 60 i 90, naravno, nisu potrebne.

```
70 LET K=IN 31: IF K < 255 THEN PRINT „EPROM NIJE PRAZAN!": GOTO MENI — pročitaj podatak sa porta A i proveri da li je to broj 255 — ako nije vrati se na „meni"
```

Programiranje

Nakon testiranja EPROM-a, program automatski prelazi na njegovo programiranje. Upisivanje sadržaja u prazan EPROM predstavlja osnovnu, i, sasvim sigurno, najdelikatniju funkciju jednog programatora. Procedura programiranja za različite tipove EPROM-a, pa čak i za 27 256, data je na dijagramima na sl. 1. modalitet „programiranje" EPROM se postavlja dovođenjem određenog napona na nožicu Vpp. Svi kontrolni ulazi nakon što se na njima uspostavi određeno logičko stanje, ostaju nakon toga mirni, osim ulaza nakon što se na njima uspostavi određeno logičko stanje, ostaju nakon toga mirni, osim ulaza PGM. Preko njega se, negativnim ili pozitivnim impulsom, upisuje podatak u EPROM. Ovaj impuls traje 50 ms. Za svaki novi bajt potreban je i nov impuls.

Program koji upravlja programiranjem nije ništa komplikovaniji od onoga koji obezbeđuje čitanje.

```
10 REM PROGRAMIRANJE
20 OUT 127, 128 — aktiviraj PIO i postavi portove
30 OUT 93, 132: OUT 93,4 — resetuj programski brojač
40 OUT 63, 94 — postavi EPROM u modalitet „programiranje"
50 FOR N=24000 TO 24000+4095 — počni od adrese 24000 a završi sa adresom 28095
60 LET K=PEEK N — uzmi podatak sa adrese N (24000)
70 IF K=255 THEN GOTO 100 — ako je na adresi N broj 255, preskoči tu memorijsku lokaciju i pređi na sledeću
80 OUT 31, K — pošalji podatak na port A
90 OUT 63, 30: FOR I=1 TO 12: NEXT I: OUT 63, 94 — generiši programski impuls u trajanju od 50 milisekundi — upiši podatak u EPROM
100 OUT 93, 5: OUT 93, 4 — pređi na sledeću adresu u EPROM-u
110 NEXT N — pređi na sledeću adresu (N+1) u RAM-u
```

Pedeset milisekundi predstavlja dovoljnu garanciju da će svaki bajt biti korektno upisan. Vreme unošenja podatka u EPROM može, međutim, biti i znatno kraće. Ali kako pronaći siguran minimum! Tako što će se odmah nakon upisa jednog podatka izvršiti provera da li je dobro upisan. „Elitniji" EPROM-i Intelove familije (2764, 27128 i 27256) u modu „verifikacija" omogućuju čitanje pod programskim naponom. To, praktično, znači da se vreme upisivanja jednog podatka može skratiti. Ako verifikacija pokaže da podatak nije upisan, upisivanje se može nastaviti. Ovu tehniku koriste tzv. „inteligentni" programatori EPROM-a u kojima programiranje traje tačno onoliko koliko je potrebno. Ova pogodnost može da bude važna kod profesionalnih uređaja, ali u amaterskoj primeni nema nikakvog značaja: „mehaničko" programiranje EPROM-a od 2 k (2716) traje manje od dva minuta, dok je EPROM-u od 16 K (27128) dovoljno dvanaestak minuta. Ova pogodnost nije ugrađena u naš programator, niti za nju postoji hardverska podrška — prvenstveno zato što u igru ulazi još jedan napon (6v)!

Opisani programator EPROM-a, međutim, ipak ne radi potpuno „mehanički": on preskače svaku lokaciju na koju treba upisati broj 255, jer se taj broj ionako nalazi na svim „praznim" adresama. O tome se brine linija 70. Lokacije FF (255) nisu, doduše, naročito česte, ali ih ponekad, kao u „galaksijinom" generatoru znakova, ima zaista puno.

Provera

Nakon programiranja, program automatski prelazi na verifikaciju, a potom, sa porukom

Nakon tri pažljivo elaborirana projekta za samogradnju u „Računarima 1“ i „Računarima 2“, hardverski nastrojeni ljubitelji računara ostali su u „Računarima 3“ gotovo bez ikakvih uzbuđenja. To , naravno, ne znači da je u redakciji preovladala softverska struja, niti da su se naši hardveristi zamorili ili izgubili smisao za konstruktorsku solidarnost. U redakciji, pored nekoliko sitnih, čekaju na red i dva veoma ozbiljna projekta — generator zvuka (potpuno završen) i fina grafika (pod malim znakom pitanja) — ali ćemo sa njima krenuti tek u „Računarima 4“ i „Računarima 5“. Zbog poznatih teškoća sa isporukom delova — koji kasne čak više i od samog QL-a (isporuka delova za memorijsko proširenje i programator eproma počinje tek u decembru!) — graditelji „galaksije“ su počeli ozbiljno da gube korak sa redakcijom. Zato, dok nas oni koji su pošli sa nama barem malo ne pristignu, ovoga puta samo završavamo ono što smo započeli u „Računarima 2“

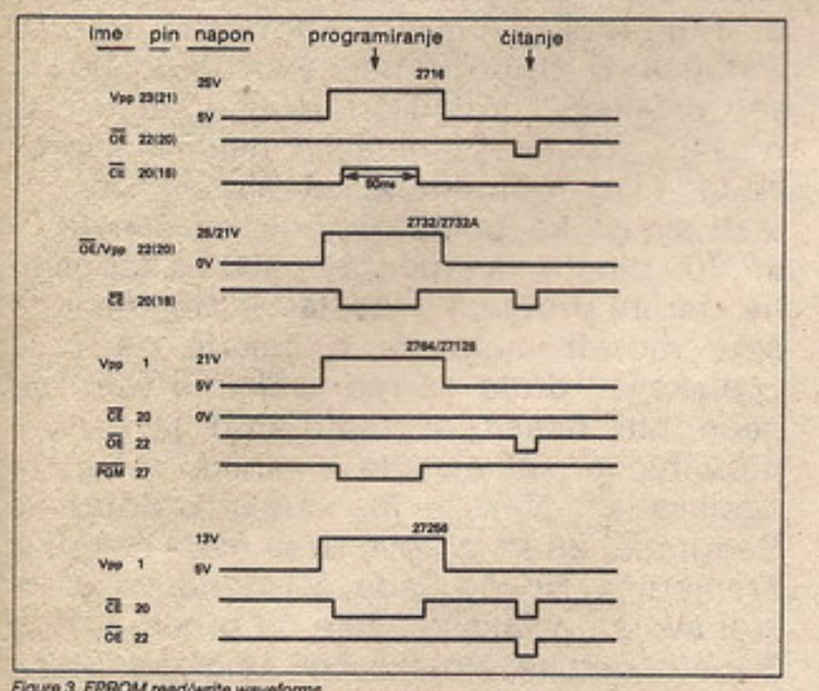


Figure 3. EPROM read/write waveforms.

Procedura programiranja i čitanja EPROM-a familije Intel: pinovi su obeleženi prema podnožju za čipove sa 28 nožica, dok je za čipove 2716 i 2732 stvarni broj nožica dat u zagradi; slika prikazuje i proceduru programiranja za EPROM-e 27256 (32 K), iako programtor ne prihvata ove tipove; dijagram daje dovoljno podataka kako se korišćenjem slobodnih C portova uređaj može adaptirati i za ovaj tip EPROM-a

„Programiranje uspelo“ ili (verovatno se nikada neće desiti) „Programiranje nije uspelo“, na „meni“. Iako čistunci priznaju samo verifikaciju bajt na bajt pod programskim naponom, klasična provera, običnim čitanjem nakon programiranja čitavog EPROM-a, nije ništa manje pouzdana. Ako neko ne može sa ovim da se složi, može lako da napiše rutinu za proveru prema sopstvenim potrebama.

```
10 REM PROVERA
20 OUT 127, 144 — aktiviraj PIO i postavi portove
(port A=čitanje)
30 OUT 93, 132: OUT 93, 4— resetuj programski
brojač
40 OUT 63, 144 — postavi EPROM u modalitet
„čitanje“
50 LET K1=K — podatak u EPROM-u mora biti
jednak sa podatkom u RAM-u
60 FOR N=24000 TO 24000+4095 počni od
adrese 24000 a završi sa adresom 28095
70 LET K=IN 31: LET K1=PEEK N — pročitaj
podatak sa porta A i „upamti“ ga u promenljivoj
K: pročitaj podatak sa adrese N i „upamti“ ga u
promenljivoj K1
80 IF K<>K1 THEN PRINT „PROGRAMIRANJE
NIJE USPELO!“: GOTO MENI — ako podaci nisu
jedanki, prekini sa proverom
90 OUT 93, 5: OUT 93, 4 — pređi na sledeću
adresu EPROM-a
100 NEXT N — pređi na sledeću adresu u RAM-u
110 OUT 63, 80 — vrati EPROM u mirno stanje
```

Opisane rutine predstavljaju okosnicu programa za pogon programatora EPROM-a. One su zajedničke za sve tipove EPROM-a sa kojima radi programator — jedino su različiti podaci u OUT instrukcijama i, naravno, podaci o kapacitetu čipa.

Parametri za različite tipove EPROM-a

tip	PORT C (adresa 63)				PORT B (adresa 93)			
	neutralno stanje	čitanje	programiranje	PGM	reset		programski brojač	
2716	65	129	7	71	132	4	5	4
2732	80	144	94	30	132	4	5	4
2732A	80	144	92	28	132	4	5	4
2764	112	176	52	20	132	4	5	4
27128	112	176	52	120	130	2	3	2

Svi kontrolni bajtovi: Kontrolni bajtovi kojima se, slanjem preko portova B (adresa 93) i C (adresa 63), obezbeđuje propisana procedura čitanja i programiranja EPROM-a

Montaža

Praktična realizacija programatora EPROM-a ne bi, za konstruktore koji su ispekli zanat na „galaksiji“, trebalo da predstavlja nikakve teškoće. Električna shema je veoma jednostavna i u njoj, i pored tri CMOS čipa, nema ni delikatnih sklopova ni delikatnih komponenata. Programator odlikuje jednostavno oživljavanje — uređaj prorađuje iz prve — i visoka pouzdanost u radu. Na njemu je isprogramirano preko pet stotina EPROM-a za računar „galaksija“. Pre prvog puštanja u rad treba, ipak, proveriti „na prazno“, merenjem napona na kontrolnim pinovima na podnožju, da li su sve funkcije ispravne i, naročito, da li su programski naponi dobri. Kod EPROM-a 2732A i 2764 oni ne smeju biti veći od 22 v jer će razoriti čip, a kod 27128 dozvoljeno je odstupanje od svega + 0,5 V! Izborom Zener dioda D3 i D4 treba dovesti napone 25/21/5 V u potrebne granice. Da bi se podešavanje olakšalo, na štampanom kolu je predviđena mogućnost da se, ako je potrebno, vežu na red i po dve diode.

Brisanje

Brisanje EPROM-a — na primer, da bi se u njega unela najnovija verzija nekog sistemskog programa — predstavlja znatno zametniji posao od programiranja — ne toliko zbog toga što duže traje koliko zato što je taj proces znatno teže kontrolisati. Brisanje EPROM-a vrši se optičkim putem (otuda i prozorče na čipu) — ultraljubičastom svetlošću talasne dužine oko 4000 mikrometara. Izvesne komponente ovog zračenja sadrže i sunčeva i fluorescentna svetlost, ali u tako malim dozama da brisanje na suncu traje nekoliko dana, a pod fluoroscentnom lampom i nekoliko — meseci.

Stoga se za brisanje koriste specijalne UV lampe talasne dužine 2537 mikrometara i energije zračenja 12mW/cm^2. Sa njima brisanje, na odstojanju od 2,5 cm!, traje 15 do 20 minuta. Ovakve lampe nije uvek lako naći. Umesto njih, u nuždi, mogu dobro da posluže i medicinske kvarcne lampe koje se koriste u kozmetici, ali one ne odgovaraju sasvim ovoj svrsi. Njihova snaga (300 i više vati) je prevelika (snažno toplotno zračenje) i zbog toga se EPROM mora udaljiti na najmanje 20 cm od svetleće cevi, i to pod uslovom da su prateći grejači demontirani (grejače treba samo ukloniti iz lampe, ali moraju ostati u kolu jer lampa neće raditi). Pošto talasna dužina svetlosti kvarcne lampe ne odgovara sasvim standardima za brisanje EPROM-a, vreme „regeneracije“ dramatično zavisi od proizvođača i tolerancije koju je ugradio u svoj proizvod. Tako je EPROM-ima firme „Hitachi“ obično dovoljno 30 minuta, „NEC“-u je potrebno oko 8 časova, a licencnim EPROM-ima manje poznatih proizvođača i čitav dan! Vreme brisanja ne zavisi od kapaciteta EPROM-a — ono je jednako i za EPROM-e od 2 i od 16 kilobajta.

Brisanje EPROM-a ne mora teći u kontinuitetu što, praktično, znači da se u programatoru može s vremena na vreme proveriti da li je posao doveden do kraja. U svakom slučaju, od nešto veće doze ne može biti nikakve štete — preporučuje se, čak da vreme brisanja bude trostruko veće od onog koje je potrebno da se obriše i poslednji bajt — pa prilikom brisanja ne treba biti preterano nežan.

Podešavanje

Programski i kontrolni napon na nožici 21 obezbeđuju tranzistori T3 do T5, kojima se upravlja preko PIO porta C, i zener diode D3 i D4. Naponi na nožici Vpp zavise, očigledno, od tolerancije zener dioda. Za napon od 21 V „uključuje“ se dioda D3 na kojoj pada napon za oko 4V, a za napon od 5 V obe diode, koje u zbiru obezbeđuju pad napona od 20 V. Na napon od 25 V ne utiče ni jedna od njih i on zavisi od kvaliteta ispravljača. Na štampanom kolu predviđena je mogućnost da se dve diode vežu na red. Tačna vrednost se može dobiti kombinovanjem više dioda manje vrednosti. U svakom slučaju, nije dozvoljeno da programski napon odstupa od predviđenog za više od ±5%, dok kod EPROM-a 27128 tolerancija iznosi samo ±0,5 V!

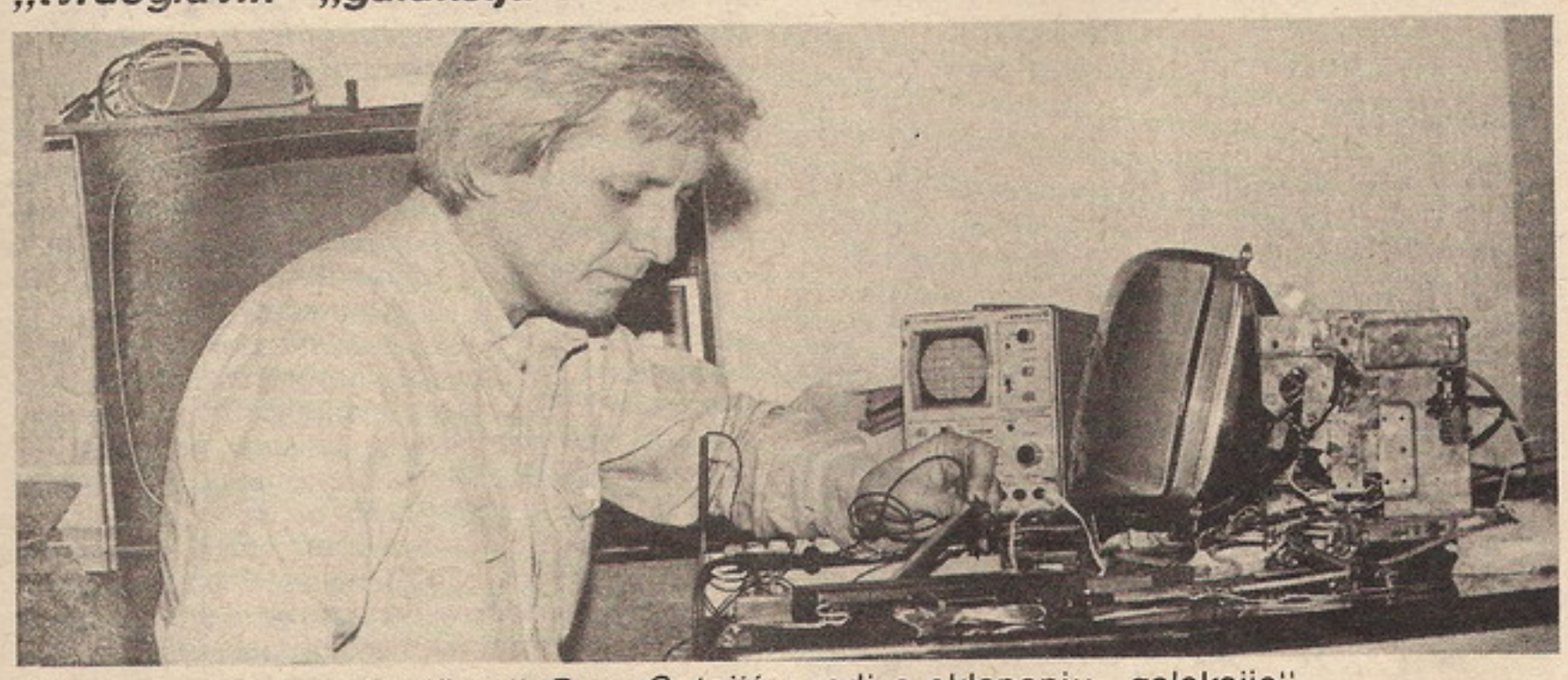
Programi u bejziku mogu korisno da posluže kao primer ne samo kako se radi sa jednim periferijskim uređajem preko ulazno izlazne memorijske mape mikroprocesora nego i za podešavanje programatora EPROM-a. U tabeli 2 su navedeni svi kontrolni bajtovi za upravljanje uređajem. Izvršavanjem jedne po jedne programske linije mogu se proveriti i podesiti sve funkcije za svaki tip EPROM-a. Osim na dijagramima na slici 1, uporedni pregled procedura dat je i u tabeli u „Računarima 2“.

Jova Regasek

„Galaktičari“ u nevolji

„ubiću se, neće da radi!“

Nakon letnje oseke sa isporukom i domaćih i inostranih delova, „galaktičari“ su s jeseni ponovo živnuli — u redakciju nedeljno stiže na desetine eproma, a poznati telefon 650-161 zvoni barem deset puta na dan da nas obavesti o još jednom srećnom sklopljenom računaru ili da, ne tako retko, zamoli za pomoć. Iako se radi o izuzetno pouzdanom projektu, redakciji se do sada obratilo stotinak čitalaca sa očekivanom dramatičnom rečenicom: „Ubiću se, neće da radi!“. U redakciji, na žalost, najčešće ne može da se dobije stručan savet, ali to ne znači da su malerozni „galaktičari“ prepušteni samima sebi. U saradnji sa radio-klubom „Avala“ iz Beograda organizovali smo besplatan servis za pregled i oživljavanje „tvrdoglavih“ „galaksija“.



Hitna pomoć za „galaktičare“: Bane Ostojić u dežurnoj hardverskoj službi

Ima nade za „galaksije“ koje „neće da rade“! Svakog utorka u prostorijama radio-kluba „Avala“, u ulici Ćirila i Metodija 2a u Beogradu, u rukama Baneta Ostojića „oživljavaju“ „galaksije“ koje iz bilo kojih razloga nisu htele da prorade u rukama svojih maleroznih vlasnika. Do sada je, zahvaljujući Banetu, popravljano na desetine ovih računara, koji su do njega stizali iz svih krajeva naše zemlje.

Hitna pomoć za konstruktore

Utorkom od 14 do 17 časova „radno mesto“ Baneta Ostojića izgleda ovako: Bane i njegov pomoćnik rade za jednim stolom, ispred njih se uvek nalaze dve-tri otvorene „galaksije“, a oko njih se obavezno muva još desetak vlasnika „galaksije“, čiji su računari trenutno na popravci ili tek treba da dođu na red. Za sve to vreme u susednoj prostoriji telefon kluba neprestano zvoni; kao što se može pretpostaviti, svi pozivi su za Baneta. On je u poslednje vreme postao popularan, ne samo kada su u pitanju opravke računara, već i kad se radi o sklapanju „galaksije“.

Bane Ostojić, inače TV-serviser u „Gorenju“, priznaje da o „galaksiji“ nije baš mnogo znao sve dok se nije pojavilo specijalno izdanje „Računari u vašoj kući“. Tada se zainteresovao za ovu akciju koja je, po njegovim rečima, „pokrenula pravu lavinu“, i odlučio da u njoj ima i sopstveni doprinos. Od aprila, kad je počeo da popravlja neispravne „galaksije“, kroz njegove ruke prošlo je više od 40 ovih računara, a za sve to vreme, Bane je neumorno delio savete vlasnicima računara preko radio-emisije Ventilator 202.

Prema Banetovim rečima, „galaksije“ najčešće boluju od neispravnog procesora. U takvim slučajevima, Bane ne može mnogo da pomogne: on samo postavlja „dijagnozu“.

„Dešava se, međutim, da konstruktori „galaksije“ ne odaberu dobru žicu za kratkospojnike, „kaže Bane“. Bolje je staviti i najobičniju bakarnu licnu za osigurače, nego eksperimentisati sa predebelim, starim ili nečistim žicama“.

Takođe se dešava da konstruktor ne ume da odabere pravu lemilicu za rad. Maksimalna snaga lemilice treba da bude 18W, ali ko već ima iskustva u lemljenju, može da radi i sa jačom.

Na spisku čekanja

Ima i banalnih kvarova: događa se da „snalažljivi“ konstruktori, u nedostatku odgovarajućih, ugrađuju otpornike i kondenzatore u vrednostima koje su drugačije od onih na šemi. Tu postoji samo jedan izuzetak:

„To je otpornik R12, koji po šemi treba da bude 390 oma, ali je iskustvo pokazalo da ga je bolje zameniti otpornikom od 1,8 kilooma. Moguće je i već postojećem otporniku dodati „na red“ 1,5 kilooma, ali bih to preporučio samo onima koji imaju spretne ruke“, kaže Bane.

Ukoliko vam se čini da ne spadate u tu kategoriju, ako nemate iskustva u lemljenju ili vam je potrebna pomoć kod sklapanja računara „galaksija“, dovoljno je da dođete u radio-klub „Avala“ i potražite Baneta Ostojića. U klubu je nedavno odlučeno da se „galaksije“ mogu popravljati samo članovima. To ne predstavlja neki veći problem; član radio-kluba „Avala“ može postati svako ko uplati simboličnu članarinu od 200 dinara za studente i 300 za zaposlene. Jedini problem predstavlja činjenica da ćete morati malo da pričekate da vaša „galaksija“ dođe na red. Čekanje vam bar neće biti dosadno; radio-klub je postao nezvanično sastajalište vlasnika računara „galaksija“. Najviše ih, naravno, dolazi iz Beograda, ali se gotovo uvek nađe neko i iz Zrenjanina, Novog Sada, Mladenovca, a oni koji svoje „galaksije“ žele da predaju lično Banetu u ruke, dolaze čak iz Makedonije, Slovenije i drugih krajeva naše zemlje.

„Od šest „galaksija“, koliko je do sada sklopljeno u klubu, sve su prordaile „iz prve“, kaže Bane“. „Znači, ne treba nikako sumnjati u konstrukciju samog računara. Što se kvarova tiče, neke smo uspevali da otklonimo za pet minuta, a neke tek posle šest časova. Pa ipak, od svih „galaksija“ koje su nam doneli, samo dve nismo uspeli da popravimo“.

Traže se dobrovoljci

Odnedavno Bane ne mora sam da izlazi na kraj sa svim „galaksijama“ koje stignu do njega. Od početka septembra u tome mu pomaže Bunduk Miloš, pomoćnik-dobrovoljac. Miloš je i sam vlasnik jedne „galaksije“ sa kojom je u početku imao problema. Pošto je čuo da u „Avali“ imaju čoveka koji takve probleme uspešno rešava, došao je u klub, učlanio se, i od tada dolazi svakog utorka da pomaže Banetu. I Miloš ima nekoliko saveta namenjenih budućim konstruktorima „galaksije“:

„Mnogi bojažljivo stavljaju čipove, koji jesu osetljivi na statički elektricitet, ali nema opasnosti da se čip ošteti ako se dodirne rukom i sl. Pre će se to dogoditi kod nepažljivog lemljenja, preteranim zagrevanjem itd. „Galaksija“ je „žilav“ računar, pa dobro podnosi zamene čipova. Naročito treba paziti na raspored nožica kod postavljanja integralnih kola; bolje je kupiti podnožja za sva integralna kola, jer ih je onda lakše proveriti“.

Nekoliko „galaksija“, mahom iz unutrašnjosti, još uvek čeka na popravku. Bane i Miloš imaju pune ruke posla, a „galaksije“ još uvek pristižu. Često se događa da njih dvojica ostanu u klubu i posle predviđenog vremena. Oni taj posao, treba i to reći, rade bez ikakve novčane nadoknade, iz čistog entuzijazma, kao što to čine i ljudi koji su započeli čitavu akciju, i zato im odajemo puno priznanje. Utoliko pre što, na žalost znamo i za suprotne primere — „domišljate“ momke koji na nevolji neukih konstruktora prave čitav kapital!

Jelena Rupnik

da li si završio „Galaksiju"?

Anketa

Do ovog trenutka, prvih dana novembra, preko 4000 ljubitelja računara dobilo je mehaničke delove za računar „galaksija". Prema našim procenama — da li smo u pravu!? — do sada je sklopljeno oko tri hiijade računara. Da bismo ostali i dalje u vezi sa svima onima koji su učestvovali u akciji „Napravi i ti. . .", pripremili smo jednu malu anketu. Rezultati koje dobijamo biće zaista dragoceni: moći ćemo, najpre, da procenimo šta je bilo dobro a šta nije u našim naporima, lakše ćemo se opredeliti kako da naš računar dalje napreduje i, što je najvažnije, steći ćemo važne pouke za neku buduću sličnu akciju.

Zato vas molimo da popunite ovaj anketni list. Na poleđini se nalaze mali oglasi koji, za većinu čitalaca, nisu „štivo" od trajne vrednosti i pošaljite ga na adresu „Galaksija" (za anketu), Bulevar vojvode Mišića 17, Beograd. Kod većine pitanja treba zaokružiti ili dopisati jedan odgovor, dok se, izuzetno, kod pitanja broj 18, 19 i 20 očekuje nekoliko odgovora.

1. Ime i prezime_____

2. Godina rođenja_____

3. Zanimanje_____

4. Adresa, mesto stanovanja_____

5. Dosadašnji kontakti sa računarima:
1. nikakvi
2. samo kroz literaturu
3. na školskom (fakultetskom kompjuteru)
4. kod prijatelja
5. posedovao sam (posedujem) sledeće džepne stone računare_____

6. profesionalno se bavim računarima

6. Za računar „galaksija" sam najpre saznao
1. iz časopisa „Galaksija"
2. iz „Računara u vašoj kući"
3. iz nekog drugog časopisa
4. iz radio (TV emisija)
5. od prijatelja

6. _____

7. Mehaničke delove sam naručio_____

a dobio _____ 1984. godine

8. Čipove sam
1. kupio od Microtehnice
2. dobio od prijatelja (rođaka iz inostranstva)
3. lično kupio u inostranstvu
4. naručio preko pošte iz inostranstva
5. kupio preko malih oglasa u zemlji

6. _____

9. Sa nabavkom pasivnih komponenata
1. nisam imao problema
2. kupio sam ih u zemlji posle duge potrage
3. naručio sam ih iz inostranstva
4. _____

10. Ispravljač sam
1. napravio samostalno
2. kupio posredstvom malih oglasa
3. naručio od stručnog lica

4. kupio u prodavnici_____
5. već posedovao

6. _____

11. Kutiju sam
1. napravio od vitroplasta

2. napravio od_____

3. kupio od_____

4. još nisam nabavio.

12. Računar „galaksija" me košta:

1. za čipove (dinara + carina)_____

2. za pasivne komponente_____dinara

3. za ispravljač_____dinara

4. za kutiju_____dinara

5. troškovi_____dinara

6. Ukupno_____dinara od toga deviznih___

13. Računar „galaksija" mi je
1. proradio „iz prve"
2. proradio posle manjih ispravki
3. proradio uz velike probleme
4. proradio uz pomoć stručnjaka
5. i dalje neispravan

14. Na sklapanje računara sam potrošio_____

___časova rada. Obzirom na pitanje 12, smatram da
1. je bilo zaista mnogo posla
2. sam radio koliko sam očekivao
3. je bilo prilično lako
4. nikada ne bih verovao da se računar ovako jednostavno sklapa

15. Dokumenatacija koja se odnosi na sklapanje računara je bila
1. preopširna
2. onakva kakva treba da bude
3. nedovoljna, ali sam se snašao
4. toliko slaba da sam tražio pomoć stručnjaka da je dopuni

16. Uputstvo za upotrebu računara je
1. na previše početničkom nivou
2. štivo zgodno za čitanje
3. dobro, ali treba da ima više detalja za bolje programere
4. dobro, ali treba da ima više primera za početnike
5. slabo i nedovoljno
6. nije vredno čitanja

17. Bejzikom računara „galaksija" sam, obzirom na to da ROM ima 4 Kb
1. više nego zadovoljan
2. zadovoljan jer nije moglo bolje
3. prilično nezadovoljan ali — „koliko para toliko muzike"
4. nezadovoljan jer je prostor neracionalno korišćen.

18. Do sada sam
1. samo probao gotove programe
2. pisao samostalno kraće programe zbog vežbe
3. pisao kraće programe koje svakodnevno koristim
4. pisao složene programe (bejzik-mašinske) za sebe i druge

19. Želeo bih da ubuduće programe
1. pišem sasvim samostalno
2. nalazim u „Galaksiji" i „Računarima u vašoj kući"
3. Kupujem preko Kataloga programa
4. kupujem preko Malih oglasa
5. kupujem u knjižarama
6. snimam sa radio-talasa

7. _____

20. Od hardverskih dodataka želeo bih da napravim
1. proširenje memorije
2. programator eproma
3. finu grafiku
4. palice za igre
5. optičku olovku

6. interfejs za_____štampač

7. _____ interfejs

8. _____

9. _____

10. Posle samogradnje „galaksije" — nikada više!

21. Od softvera bih posebno želeo da nabavim
1. asembler
2. monitor
3. tekst procesor
4. prošireni bejzik
5. razne uslužne programe
6. mnogo, mnogo igara

7. _____

8. _____

22. Kada sve saberem i oduzmem
1. smatram da je samogradnja „galaksije" bilo divno iskustvo koje bi vredelo ponoviti
2. srećan sam što sam došao do računara, ali nije lako nešto napraviti samostalno
3. računar mi i nije bio potreban ali volim da pravim uređaje svojim rukama
4. nije trebalo da se upuštam u sve ovo, ali bih se ponovo zaleteo
5. nikada više samogradnja — kupovaću samo gotove kompjutere
6. sreća u nesreći je što me saznanje da me računari ne interesuju nije preskupo koštalo

Ostatak prostora (po potrebi još neki list) upotrebite na komentare čitave akcije, računara „galaksija", i, naročito, teškoće koje ste imali sa njegovim oživljavanjem. Na osnovu vaših zapažanja izvršićemo tipizaciju grešaka.

MZ-731

Mercator – Mednarodna trgovina

TOZD CONTAL

predstavlja vam SHARP kolekciju kalkulatora i kompjutera za sva vremena!

Iz serije 700, lični kompjuter MZ 731 sa printerom i kasetofonom — zabava—učenje—potreba
cena: 1 128 DM

uz MZ 731 možete posebno naručiti:
„Zeleni" monitor po ceni: 420 DM
ili TV kolor monitor po ceni: 840 DM
Dinarske dažbine cca 65%

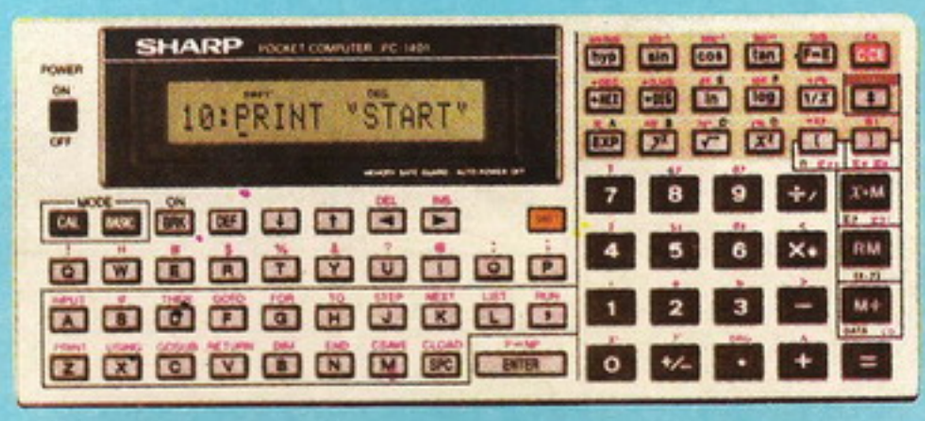
CE-152

PC-1500 A

Za poslovne ljude, inženjere, profesionalce i studente izbor iz familije džepnih kompjutera SHARP:

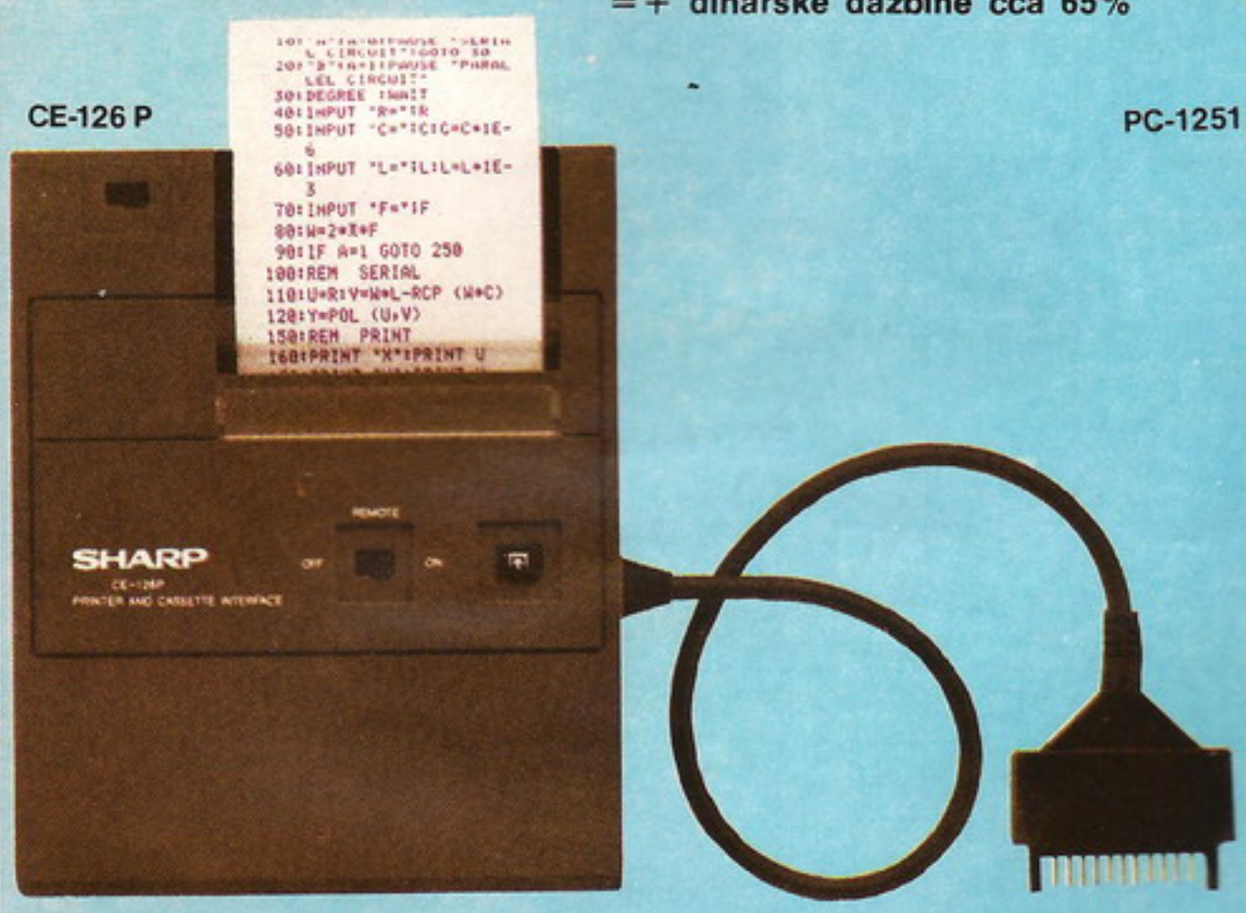
Kompjuter PC 1500 A cena: 440 DM
Kompjuter PC 1251 cena: 220 DM
Kompjuter PC 1401 cena: 230 DM
Štampač CE 126 P cena: 170 DM
Data recorder CE 152 cena: 120 DM

=+ dinarske dažbine cca 65%



PC-1401

CE-126 P



PC-1251

EL-2607

Kalkulatori s trakom:

EL-1611

EL 1611/cena 80 DM
EL 2607/cena 170 DM

+dinarske dažbine cca 65%

SHARP

ZASTUPA I PRODAJE:

TOZD CONTAL

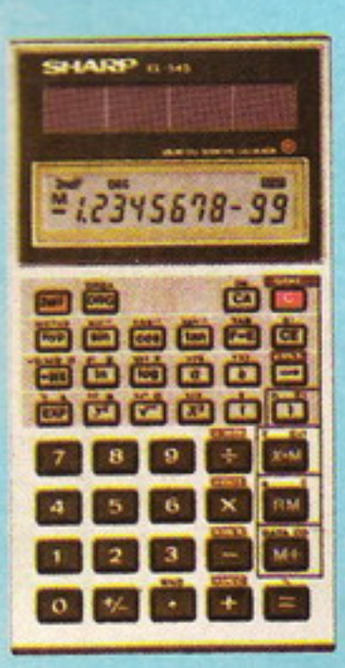
Titova 66
61000 Ljubljana
Telex: 31175
informacije na telefon: 061/328-441

„SHARP" džepni kalkulatori:

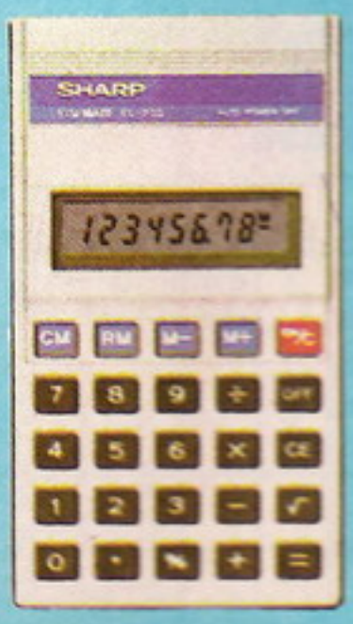
EL 230 (13 DM), solarni kalkulatori EL 240 (18 DM) i EL 862 (27 DM)
EL 863 (27 DM), profesionalni kalkulatori EL 531 (30 DM) i EL 545 (62 DM)

+dinarske dažbine za džepne kalkulatore cca 30%

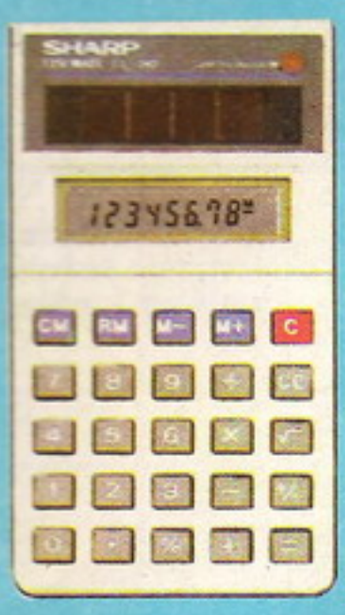
EL-531



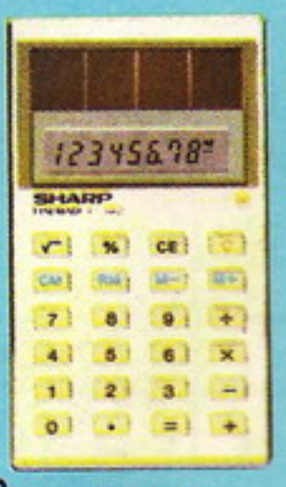
EL-545



EL-230



EL-240



EL-862

EL-863